ANALOG DEVICES

ADuCM350 Hardware Reference Manual

UG-587

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Scalable, Connected Precision Meter Platform with Intelligence

SCOPE

This reference manual provides a detailed description of the ADuCM350 functionality and features.

Disclaimer

FUNCTIONAL BLOCK DIAGRAM

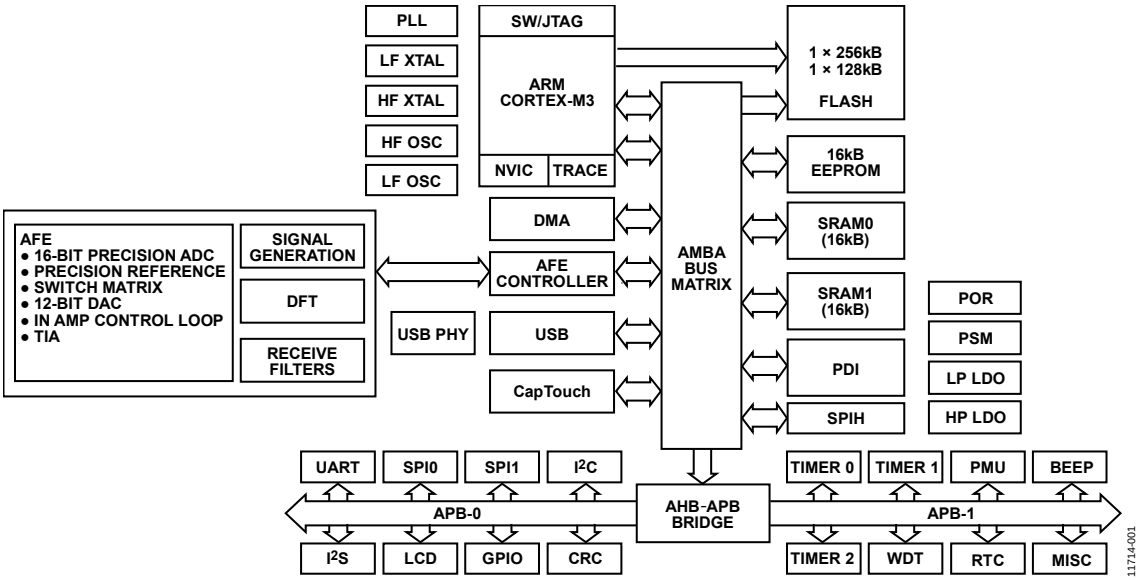


Figure 1. Functional Block Diagram

TABLE OF CONTENTS

REVISION HISTORY

1/2018—Rev. D to Rev. E

Added Limitations on Use and Liability Section 460

5/2017—Rev. C to Rev. D

Changes to Flash Integrity, Parity Feature for User Space 0 Section and Flash Integrity, Parity Feature for User Space 1 Section 140

5/2016—Rev. B to Rev. C

Changes to AFE Retention In Hibernate Section 315
Deleted Table 470; Renumbered Sequentially 315

9/2015—Rev. A to Rev. B

Changes to Figure 1 1
Changes to System Features Section 7
Changes to Debug Protocol Options Section 13
Changes to Trace Output Section 14
Changes to Clock Dividers Section 17
Changes to PLL Programming Section, Table 8, and Interrupt Section 18
Changes to Sequence Section 19
Changes to Platform LF RC Oscillator Section 20
Changes to Table 17 35
Changes to Clock Select Section and Table 31 56
Changes to Introduction Section 82
Changes to Table 102 88
Changes to Figure 34 93
Changes to Table 108 93
Changes to Figure 39 and Figure 40 Caption 96
Changes to Figure 44 and Figure 45 99
Changes to Figure 46 Caption and Figure 47 100
Changes to Table 121 111
Changes to Auto Request Section 118
Changes to Address Calculation Section 125
Changes to Flash Integrity, Signature Feature Section 139
Changes to Flash Integrity, Parity Feature for User Space 0 Section and Parity Error Interrupt Section 140
Changes to RNG Oscillator Counter Section 168
Changes to Stall Issued to Control Transfer Section 177
Changes to LPM Interrupt Enable Register Section 210
Changes to Table 309 216
Added Drive Strength Section 222
Changes to GPIO Port 1 Configuration Register Section 227
Changes to GPIO Port 1 Drive Strength Select Register Section and GPIO Port 2 Configuration Register Section 230
Changes to GPIO Port 3 Configuration Register Section 233
Changes to GPIO Port 4 Configuration Register Section 237
Changes to Functionality Section and Rx/Tx Data FIFOs Section 240

Changes to Figure 72 241
Changes to Figure 73 242
Changes to Power-Down Considerations Section 243
Changes to Table 386 244
Changes to Table 395 and Table 396 248
Changes to Table 407 252
Changes to SPI Transfer Initiation Section and Figure 74 254
Changes to Figure 75 255
Changes to DMA Master Transmit Configuration Section 256
Changes to DMA Master Receive Configuration Section 257
Changes to LCD Charge Pump and External Capacitor Requirements Section and Figure 81 279
Added Table 450 Title 287
Added Table 454 Title, Table 456 Title, and Figure 84; Renumbered Sequentially 288
Changes to Basic Loop Theory Section 300
Added How to Use the TIA to Set the AFE Loop Common Mode Section 301
Changes to Figure 97 309
Changes to Figure 98 Caption 312
Changes to Figure 100 313
Added AFE Retention in Hibernate Section and Table 470 ... 315
Changes to Waveform Generator Section 316
Changes to Trapezoid Generator Section 318
Changes to Figure 113 326
Changes to Table 489 341
Changes to No Factory Calibration Sequence Section 364
Changes to Figure 133 376
Deleted DAC Usage Cases Section 407
Changes to ADC Measurement Section 396
Changes to REF_EXCITE Functionality Section 398
Changes to Temperature Sensor Usage Section 399
Changes to Baseline Calibration Section 411
Changes to Table 565 415
Changes to Table 566 416
Changes to Table 568 and Table 569 418
Changes to Table 571 and Table 582 420
Changes to Table 584 to Table 638 424

4/2014—Rev. 0 to Rev. A

Changes to Bit 9 in Table 13 31

3/2014—Revision 0: Initial Version

INTRODUCTION

SYSTEM FEATURES

The ADuCM350 is an ultralow power, integrated mixed-signal metering solution that includes a microcontroller subsystem for processing, control, and connectivity. The microcontroller subsystem is based on an ARM Cortex™-M3 processor, a collection of digital peripherals, embedded SRAM and flash memory, and an analog subsystem that provides clocking, reset, and power management capabilities.

System features include the following:

- 16 MHz ARM Cortex-M3 processor
- 384 kB of embedded flash memory
- 32 kB system SRAM
- 16 kB EEPROM
- Integrated full speed USB 2.0 controller and PHY
- Power management unit (PMU)
- Multilayer advanced microcontroller bus architecture (AMBA) bus matrix
- Central direct memory access (DMA) controller
- I^2S and beeper interfaces
- LCD controller functions
- Serial peripheral interface (SPI), I^2C, and UART peripheral interfaces
- A real-time clock (RTC)
- An analog front-end (AFE) controller
- General-purpose, wake-up, and watchdog timers
- Programmable general-purpose inputs/outputs (GPIOs)
- A power-on reset (POR) feature and power supply monitor (PSM)
- A discrete Fourier transform (DFT) engine
- Receive filters
- Six-button CapTouch® interface
- 12-bit digital-to-analog converter (DAC)
- Temperature sensor
- Instrumentation amplifier control loop
- 16-bit analog-to-digital converter (ADC) performance
- High precision voltage reference

To support extremely low dynamic and standby power management, the ADuCM350 provides a collection of power modes and features, such as dynamic and software controlled clock gating and power gating.

The AFE is connected to the microcontroller subsystem via an advanced high performance bus (AHB) slave interface on the AMBA bus matrix, as well as via the DMA and interrupt connections.

Note that throughout this hardware reference manual, multifunction pins (such as the TMS-SWDIO/P0.8 pin) are referred to either by the entire pin name or by a single function of the pin (for example, TMS-SWDIO) when only that function is relevant.

Full specifications on the ADuCM350 are available in the product data sheet, which should be consulted in conjunction with this reference manual when working with the device.

SYSTEM BLOCK DIAGRAM

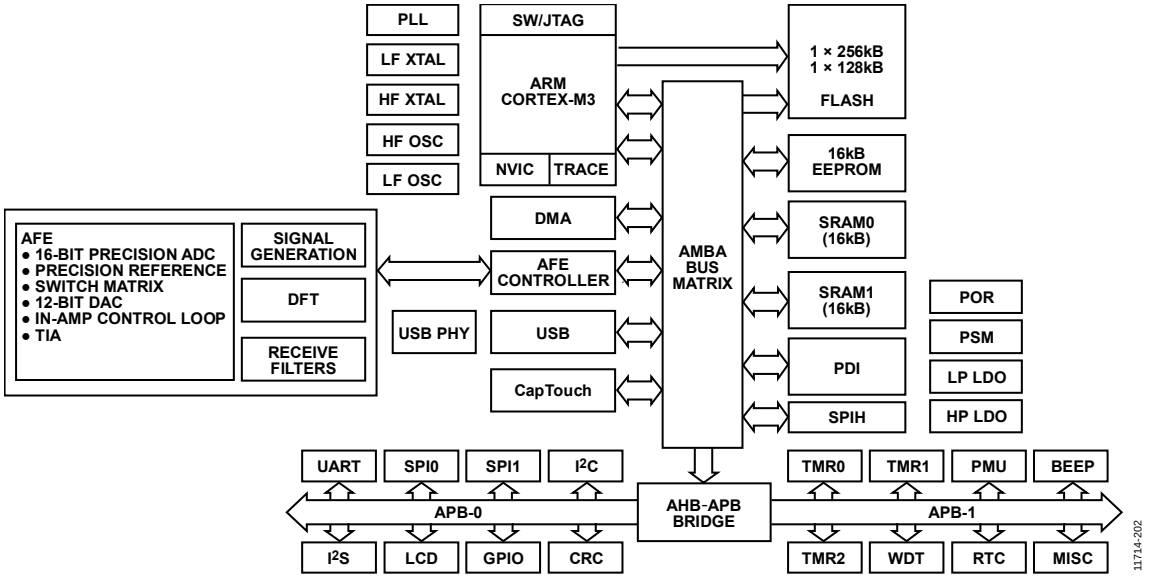


Figure 2. System Block Diagram

The ARM Cortex-M3 is a powerful 32-bit reduced instruction set computer (RISC) processor, offering up to 20 MIPS peak performance at 16 MHz. A central DMA controller is used to efficiently move data between peripherals and memory. On-chip 384 kB of nonvolatile flash memory, 16 kB EEPROM, and 32 kB of SRAM are also provided.

The device provides a range of flexible clocking features, allowing the system to operate from an internal RC oscillator, a crystal oscillator, or a phase-locked loop (PLL). A range of programmable clock divisions allows software to run the system at the minimum clock frequency required to save power.

The system also integrates a range of on-chip peripherals that can be configured via the microcontroller software for a given application. These peripherals include USB, UART, I^2C, I^2S, and SPI communication controllers, a GPIO port, general-purpose timers, a wake-up timer, and a system watchdog timer.

A PMU provides the system with four power modes that enable various levels of clock and power gating to minimize power consumption. In hibernate mode, activity on peripheral interfaces or events firing from the wake-up timer can wake up the device. Both software controlled and automatic clock gating are used to reduce dynamic power consumption, and power gating is also used to reduce standby current.

The AFE comprises a 12-bit DAC, an ADC with 16-bit performance, an instrumentation amplifier control loop, an integrated temperature sensor, a precision reference, and a pin switch matrix. The waveform generator DFT engine, receive filters, and AFE controller decouple the microcontroller core from the core analog control functions, which results in a highly tailored, power efficient measurement engine.

A six-button CapTouch interface allows the ADuCM350 to interface via a capacitive sensor using self capacitance measurements. The CapTouch interface supports a low power proximity mode; it also incorporates various noise reduction techniques, making it robust to environmental conditions.

MEMORY ORGANIZATION

The ADuCM350 incorporates 384 kB of embedded flash memory for program code and nonvolatile data storage, 16 kB of EEPROM, and 32 kB of SRAM. The flash memory uses 2 kB of information space (not shown in the memory map), which includes a software kernel as well as manufacturing and test data.

Table 1 shows the ADuCM350 address map. Access to undefined address regions result in a memory management exception.

The SRAM and peripheral memory mapped registers are mapped to both the bit band region as well as a bit band alias region. The bit band alias region allows individual bits in memory to be accessed as words, allowing bit operations without having to write bit manipulation routines or additional language extensions.

The internal and external private peripheral bus regions provide access to M3 modules (if they are configured as being present), such as nested vectored interrupt controller (NVIC), instruction trace module (ITM), or flash patch and breakpoint (FPB) unit. For more information about these peripheral bus modules, consult the ARM Cortex-M3 Technical Reference Manual.

Table 1. ADuCM350 Memory Map

First Address	Last Address	Memory Region
0x0000_0000	0x0005_FFFC	384 kB flash memory
0x2000_0000	0x2000_3FFF	RAM Bank 0 (16 kB)
0x2004_0000	0x2004_3FFF	RAM Bank 1 (16 kB)
0x2008_0000	0x2008_3FFF	16 kB EEPROM
0x4000_0000	0x4000_001C	General-Purpose Timer 0
0x4000_0400	0x4000_041C	General-Purpose Timer 1
0x4000_0800	0x4000_081C	General-Purpose Timer 2
0x4000_2400	0x4000_2404	System power control
0x4000_2420	0x4000_2434	External interrupts
0x4000_2440	0x4000_2440	System reset
0x4000_2500	0x4000_2540	Wake-up timer
0x4000_2580	0x4000_2598	Watchdog timer
0x4000_2600	0x4000_2618	Real-time clock
0x4000_3000	0x4000_3050	I^2C master/slave
0x4000_4000	0x4000_4018	SPI0 master/slave
0x4000_4400	0x4000_4418	SPI1 master/slave
0x4000_5000	0x4000_5030	UART
0x4000_5800	0x4000_5820	I^2S master/slave
0x4000_5C00	0x4000_5C08	Beeper
0x4000_6000	0x4000_6010	Random bit generator
0x4000_8000	0x4000_804C	LCD controller
0x4001_0000	0x4001_0FFC	DMA
0x4001_8000	0x4001_807C	Instruction flash controller
0x4001_C000	0x4001_C07C	General-purpose flash controller
0x4002_0000	0x4002_00B4	GPIO
0x4002_4000	0x4002_4018	SPIH master/slave
0x4002_C000	0x4002_C032	CRC engine
0x4003_0000	0x4003_0048	Parallel display interface
0x4008_0000	0x4008_0144	AFE
0x400A_0000	0x400A_03B4	USB controller

IDENTIFICATION REGISTERS

The identification registers are described in Table 2.

Table 2. Identification Registers Summary

Address	Name	Description	Reset	RW
0x40002020	ADIID	Analog Devices identification register	0x4144	R
0x40002024	CHIPID	Chip identification register	0x0210	R

Analog Devices Identification Register

The ADIID identification register is a 16-bit field that is present on all implementations of the Cortex-M3 low power platform. It is designed to be used by debuggers to confirm that the device that they are connected to via a serial wire is an implementation of the Cortex-M3 low power platform from Analog Devices. Debuggers can subsequently check the CHIPID identification register to identify the particular implementation used. The ADIID register has a default value of 0x4144, which does not vary between implementations. The ADIID register is located at Address 0x40002020. This address and value have been notified to tool vendors for device identification purposes.

ADIID Register

Address: 0x40002020, Reset: 0x4144, Name: ADIID

Table 3. Bit Descriptions for ADIID

Bits	Bit Name	Description	Reset	Access
[15:0]	ID	The fixed value of 0x4144 is present at this register.	0x4144	R

Chip Identification Register

The CHIPID identification register is a 16-bit field that is unique to an implementation of this Cortex-M3 low power platform. It can be used by code running on the Cortex-M3 to tailor its operation for that implementation. It is also designed for use by the serial wire debug tools to allow external debuggers to check the particular implementation of the Cortex-M3 low power platform and to tailor its operation for that implementation. The CHIPID register is located at Address 0x40002024.

CHIPID Register

Address: 0x40002024, Reset: 0x0210, Name: CHIPID

Table 4. Bit Descriptions for CHIPID

Bits	Bit Name	Description	Reset	Access
[15:4]	PARTID	Part identifier	0x21	R
[3:0]	REVISION	Silicon revision number	0x0	R

ARM CORTEX-M3 CORE

FEATURES

The ADuCM350 contains an embedded ARM Cortex-M3 processor. The Cortex-M3 processor is an ARM processor for low cost embedded systems targeted at ultralow power processing requirements.

The ARM Cortex-M3 32-bit RISC processor features exceptional code efficiency, delivering the high performance expected from an ARM core in the memory footprint typically associated with 8-bit and 16-bit devices.

High Performance

High performance features include the following:

- 1.25 DMIPS/MHz.
- Many instructions, including multiply are single cycle.
- Separate data and instruction buses allow simultaneous data and instruction accesses to be performed.
- Optimized for single cycle flash usage.

Low Power

Low power features include the following:

- Low standby current. The device includes a low power wake-up timer.
- Core implemented using advanced clock gating; therefore, only actively used logic consumes dynamic power.
- Power-saving mode support (sleeping and deep sleep). The design has separate clocks to allow unused parts of the core to be stopped.

Advanced Interrupt Handling

The NVIC supports up to 240 interrupts. The vectored interrupt feature greatly reduces interrupt latency because there is no need for software to determine which interrupt handler to serve. In addition, there is no need to have software to set up nested interrupt support.

The ARM Cortex-M3 processor automatically pushes registers onto the stack at interrupt entry and pops them back at interrupt exit. This reduces interrupt handling latency and allows interrupt handlers to be normal C functions.

Dynamic Priority Control for Each Interrupt

Dynamic priority control for each interrupt includes the following:

- Latency reduction using late arrival interrupt acceptance and tail chain interrupt entry
- Immediate execution of a nonmaskable interrupt request for safety critical applications

System Features

System features include the following:

- Support for bit band operation, byte invariant big endian mode and unaligned data access.
- Advanced fault handling features include various exception types and fault status registers.

Debug Support

Debug support includes the following:

- Serial wire and JTAG debug interfaces (SW-DP)
- Embedded trace Macrocell™ provides high bandwidth instruction trace
- Flash patch and breakpoint (FPB) unit for implementing breakpoints
- Data watchpoint and trigger (DWT) unit for implementing watchpoints, trigger resources, and system profiling

FUNCTIONAL DESCRIPTION

The ARM Cortex-M3 Technical Reference Manual describes all possible features of an ARM Cortex-M3 in detail.

As noted in the ARM Cortex-M3 Technical Reference Manual, several Cortex-M3 components are flexible in their implementation, such as SW/JTAG-DP, ETM, TPIU, ROM table, MPU, and NVIC.

Serial Wire Debug (SW/JTAG-DP)

The device supports both the serial wire and JTAG interfaces via a common set of pins that multiplex both functions. See the ARM CoreSight™ Components Technical Reference Manual for details on SWJ-DP.

ROM Table

The default ROM table was implemented as described in the ARM Cortex-M3 Technical Reference Manual.

NVIC

The ARM Cortex-M3 processor includes an interrupt controller, NVIC. It is closely coupled with the processor core and provides several features that include the following:

- Nested interrupt support
- Vectored interrupt support
- Dynamic priority changes support
- Interrupt masking

In addition, the NVIC also has a nonmaskable interrupt (NMI) input. The NVIC is implemented in the ADuCM350 and is described in more detail in the Nested Vectored Interrupt Controller section.

Wake-Up Interrupt Controller (WIC)

Analog Devices has a modified WIC to provide the lowest power-down by removing the need for a clock to wake up the device. For more information, see the System Control Register section.

DMA

The ADuCM350 implements the ARM µDMA. For more information, see the DMA Controller section.

For more information about the ARM Cortex-M3 processor core, see the following:

- ARM Cortex-M3 Technical Reference Manual
- ARM Cortex-M3 Devices Generic User Guide
- ARM Cortex-M3 Errata

DEBUG INTERFACE

INTRODUCTION

The ARM Cortex-M3 processor (R2P1) on the ADuCM350 supports two types of debug host interfaces, a 4-wire JTAG debug (JTAG) interface and a 2-wire serial wire debug (SWD) interface. To make efficient use of the package pins, the serial wire interface shares its pins with the JTAG interface.

Table 5. Serial Wire and JTAG Pins

CSPBGA Bump Location	Package Pin	JTAG	Serial Wire
H1	TMS-SWDIO	TMS	SWDIO
H2	TCK-SWCLK	TCK	SWCLK
J2	TDI	TDI	Not applicable
J1	TDO-SWO	TDO	Not applicable

DEBUG PROTOCOL OPTIONS

The two protocols, JTAG (IEEE 1149.1 Standard Test Access Port) and serial wire debug, are supported on the ADuCM350 for debug. Upon a power-on reset, the JTAG debug interface is selected by default. The user has the ability to switch between the SWD and JTAG by sending a specific sequence of 16 bits on the TMS-SWDIO pin. See the ARM Debug Interface v5 supplemental document for more details about this switching.

INTERFACE ACCESS

The debug interface signals are selected by default on the pins. The GPIO pins can be assigned alternate function using the GPxCON port configuration registers. Care must be taken when selecting an alternate function because the debug interface is then not available to the user. In addition, the user must consider the state of DBG (FEECON1 Flash Control 1 register). If DBG is disabled, the debug of user code is prevented. Ensure that GPIO configuration, as well as enabling the debug interface, are part of the code debug strategy.

DEBUG FEATURES SUPPORTED

The processor contains several system debug components that facilitate low cost debug, trace and profiling, breakpoints, watchpoints, and code patching, the details of which can be obtained from the ARM Cortex-M3 Technical Reference Manual.

The supported system debug components are as follows:

- Flash patch and breakpoint (FPB) unit to implement breakpoints and code patches.
- Data watchpoint and trace (DWT) unit to implement watchpoints, trigger resources, and system profiling.
- Instrumentation trace macrocell (ITM) for application driven trace source that supports printf style debugging.
- Embedded trace macrocell (ETM) for instruction trace. This is an optional feature that may be supported.

The details and programming model for all of these features can be obtained from the ARM Cortex-M3 Technical Reference Manual.

EMBEDDED TRACE

FEATURES

The ADuCM350 incorporates the complete embedded trace of the ARM Cortex-M3 processor features to maximize the code analysis, system profiling, and debugging capabilities. Data tracing, watchpoints, and various system profiling features are available from the data watchpoint and trace (DWT) unit. The instrumentation trace module (ITM) provides application-driven trace source that supports printf style debugging and generates system diagnostic information. The embedded trace macrocell (ETM) provides real-time instruction trace information.

TRACE IMPLEMENTATION

The ADuCM350 implements the ARM ETM architecture v3.5. See the ARM Embedded Trace Macrocell Architecture Specification for detailed architecture and usage information.

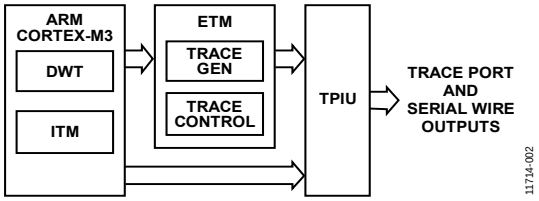


Figure 3. Embedded Trace Block Diagram

The DWT provides a selection of debugging features and contains four comparators that can each be programmed to be a hardware watchpoint, an ETM trigger, a PC sampler event trigger, or a data address sampler event trigger. It also includes several counters that can be used to record how many clock cycles or instructions various functions take to complete. The DWT is described in detail in the ARM Cortex-M3 Technical Reference Manual.

The ITM allows application software the ability to write console messages and output these messages as trace data, generating trace information as packets to the trace port interface unit (TPIU). It also generates and inserts timestamps into the trace stream to help the debugger to identify event ordering. The ITM is described in both the ARM Cortex-M3 Technical Reference Manual as well as the ARMv7-M Architectural Reference Manual.

The ETM generates instruction trace packets to be transmitted from the trace port interface at the core clock rate to the off chip debugger. Both 16-bit and 32-bit thumb instructions can be traced. Tracing is enabled and disabled by using four compare registers in the DWT, by using up to two external trace sources inside the Cortex-M3, or by using dedicated start and stop bits controlled by the DWT. A 24-byte hardware first in, first out (FIFO) is used to buffer instruction packets.

The TPIU provides the interface between the ITM and ETM modules, formatting and serializing the packets before transmission out the trace port to an off chip trace port analyzer.

TRACE OUTPUT

There are two formats of trace data output that are supported on the ADuCM350. One format is the 4-pin parallel trace, which uses the standard 4-pin version of the Cortex-M3 trace port interface unit.

For reduced pin count, the single pin trace output option is also available. Serial wire trace output that is multiplexed with TDO-SWO pin can be used to output trace data. However, this is not recommended to be used along with ETM. More details can be obtained from the ARM Cortex-M3 Technical Reference Manual.

SYSTEM CLOCKS

FEATURES

The ADuCM350 integrates two on-chip oscillators and the circuitry for two external crystals:

- LFOSC is a 32 kHz internal oscillator.
- HFOSC is a 16 MHz internal oscillator.
- LFXTAL is a 32 kHz external crystal oscillator.
- HFXTAL is an 8 MHz or 16 MHz external crystal oscillator.

A duty cycle correction (DCC) block is used to balance the output clock from HFXTAL to 50% duty cycle for the system. It is enabled together with HFXTAL.

Two on-chip PLLs are also available: the system PLL (SPLL) and the USB PLL (UPLL). Both PLLs can use either the HFOSC or the HFXTAL as an input clock.

The high frequency oscillators (HFOSC and HFXTAL), along with the output of the SPLL, can be used to generate the root clock.

The root clock is divided into many internal clocks:

- FCLK clocks the NVIC, which includes SYSTICK of the Cortex-M3 core.
- HCLK_CORE clocks the Cortex-M3 core.
- HCLK_BUS clocks the advanced high performance bus (AHB) peripherals.
- PCLK clocks the advanced peripheral bus (APB) peripherals.
- Each peripheral has a UCLK that is synchronous to the PCLK for its internal operation (with the exception of SPIH, which is synchronous to HCLK). Each UCLK can be gated individually.
- USBCTLCLK clocks the USB control logic and must be ≥30 MHz when the USB is connected.
- ACLK clocks the AFE and must be 16 MHz when the analog front end is actively working.
- AFE_ADC_CLK clocks the ADC inside AFE and must be 16 MHz when the analog front end is actively working. AFE_ADC_CLK shares the same control bits as ACLK: CLKCON5[7] (ACLKOFF).
- HCLK_CT clocks all the digital in the CapTouch block except for the ADC controller and the self timer. HCLK_CT shares the same control bit in Power Mode 2: CLKCON5[8] (CTCLKOFF). This clock must be 16 MHz when the CapTouch is actively working.
- CTCLK clocks the ADC controller only, and generates clocks and control signals for the CapTouch analogue section and must be 16 MHz when the CapTouch is actively working.

A USBPHYCLK for clocking the USB PHY is also available and must use a 60 MHz clock when the USB is connected; however, its source is the direct output of the UPLL, not the root clock.

The 32 kHz clock (LF_CLK) is generated from either LFXTAL or LFOSC to drive certain blocks, including the watchdog timer, LCD, beeper, and CapTouch self timer. The wake-up timer and general-purpose timers have their own multiplexers to select their clock source. The RTC always works from LFXTAL.

Figure 4 shows a diagram of all the clocks available and includes the clock gates for power management.

Note that the delay block (DLY) is used to offset the clock phase of the digital system away from the phase of the AFE and CapTouch ADCs to ensure that the ADCs can sample correctly and that the digital system is as quiet as possible.

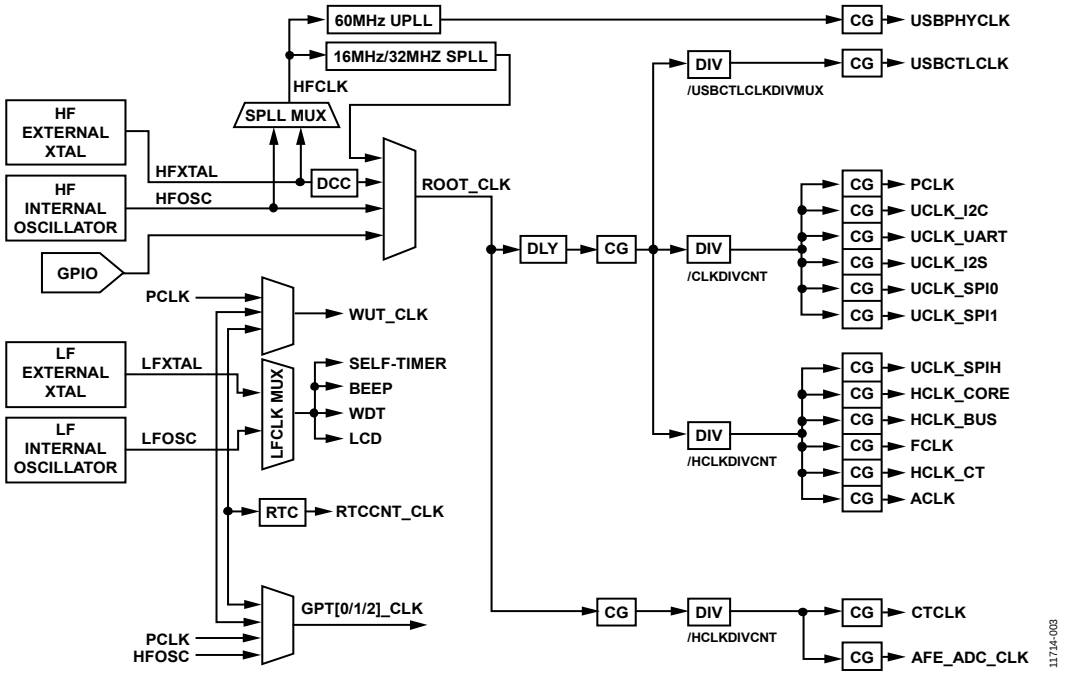


Figure 4. Clocking Diagram

OPERATION

At power-up, the core executes from a 1 MHz system clock. The user is then given control to program a clocking scheme appropriate for the application, which is done by setting values in the clocking registers.

Clock Muxes

There are three clock source multiplexers, as shown in Figure 4: the root clock mux (ROOT_CLK), the low frequency clock mux (LFCLK Mux) band, and the SPLL input mux (SPLL Mux). These clock source multiplexers are controlled using memory mapped registers.

The dashed multiplexers for the wake-up timer and general-purpose timers are controlled via registers within the corresponding blocks. See the Wake-Up Timer (WUT) section and the General-Purpose Timers section for details.

In general, ensure that the desired clock input is available and stable for the clock selection made for the clock mux. Otherwise, situations where the system can be locked out of a stable clock can arise. Registers used to program the clock settings are on the HCLK_BUS domain.

The ADuCM350 provides the facility to observe the various clocks used in the system on an external pin. The GPIO Pin P1.7 is used for this purpose. This feature is available as Function 4 on the P1.7 pin. To observe this clock on Pin P1.7, select the function using the GP1CON register (GP1CON[15:14] = 0x3). The clock source that appears on Pin P1.7 is selected in the miscellaneous clock settings register (CLKCON0[7:4]).

As an example, to view the LFXTAL clock on the P1.7 pin, the following needs to be selected:

- The GPIO clock source needs to be selected in Bits[1:0] of the CLKCON0 register.
- The LFXTAL must be enabled in Bit 8 of the CLKCON0 register.

Table 6. Clock Muxing

Clocks	Register	Bit(s)	Selection
Root Clock Mux	CLKCON0	[1:0]	00: HFOSC 01: HFXTAL 10: SPLL 11: GPIO (Pin P0.11 test purposes only)
SPLL Mux	CLKCON0	11	0: HFOSC 1: HFXTAL
LFCLK Mux	CLKCON0	8	0: LFOSC 1: LFXTAL
WUT_CLK Mux	T2CON	[10:9]	Controlled by the wake-up timer
GPTX_CLK Mux	GPTXCON	[6:5]	Controlled by General-Purpose Timer 0/General-Purpose Timer 1/General-Purpose Timer 2

Clocks	Register	Bit(s)	Selection
GPIO Out Clock	CLKCON0	[7:4]	0000: ROOT_CLK 0001: LF_CLK 0010: CTCLK 0011: HCLK_BUS 0100: HCLK_CORE 0101: PCLK 0110: USBCTRLCLK 0111: USBPHYCLK 1000: GPT0_CLK 1001: GPT1_CLK 1010: GPT2_CLK 1011: WUT_CLK 1100: RTCCNT_CLK 1100 to 1111: not used

Clock Dividers

Four programmable clock dividers are available to generate the clocks in the system. A clock divider integer divides the input clock down to a new clock. The division range is from 1 to 32. Division selection can be made on the fly. The output remains glitch free and stretches the high time, never creating a high time shorter than the pre or post value of the clock period.

Three of the clock dividers use the root clock as an input and generate the core and peripheral synchronized clocks. The clock dividers are cascaded in such a way that each stage initially releases its divided clock in a sequence. The last stage informs all other stages that its divided clock is ready to be output. The effect of this cascading is that divided clocks are released synchronously when new divider values are programmed. Initial edges of each clock are mutually aligned.

An additional independent divider is available to generate the CapTouch clock. It uses the same divider counter as HCLKDIVCNT because CTCLK always has the frequency of HCLKs. Table 7 summarizes the inputs and outputs of each clock divider along with the register bits to program each clock divider.

Table 7. Clock Dividers Sources and Outputs

Divider	Input Clock	Output Clocks	Bits in the CLKCON1 Register
0	Root clock	HCLK_CORE, HCLK_BUS, FCLK, ACLK, CTCLK, AFE_ADC_CLK	HCLKDIVCNT
1	Root clock	PCLK, all peripheral UCLKs	PCLKDIVCNT
2	Root clock	USBCTLCLK	USBCTLCLKDIVMUX

Divider 2 does not have a programmable divide value. Its divide value can be either HCLKDIVCNT or half of HCLKDIVCNT. When the USBCTLCLKDIVMUX bit is set to 0, USBCTLCLK equals HCLK. When the bit is set to 1, the USBCTLCLK is twice the frequency of HCLK. Note that for the 2× relationships to hold between HCLK and USBCTLCLK, the HCLK divider must be an even number. If the LSB of the divider is 1, it truncates when the divider value is shifted right by 1. For example, if the HCLK divider is 7 and USBCTLCLKDIVMUX is 1, the USBCTLCLK divider is 3, making the illegal relationship between the two clocks 7/3 instead of 2/1.

Only certain divider ratios are legal between PCLK and HCLK. Specifically, the following rules must be respected:

- The frequency of PCLK must always be smaller or equal to the frequency of HCLK.
- The ratio of the dividers must be an integer.
- PCLK must be the same speed as HCLK before going into hibernate mode.

In general, clock division can be changed on the fly during normal operation; however, one exception is a USB transfer. Do not distribute clocks during a USB transfer; therefore, do not program a new divider until USB tasks are completed.

Clock Gating

In the case of certain clocks, clocks can be individually gated depending on the power mode or register settings. For more information about clock gating and power modes, refer to the Power Management Unit section.

The clock gates of the peripheral UCLKs are user controllable in certain power modes. Register CLKCON5 can be programmed to turn off certain clocks, depending on user application. Set the respective bits to 1 in the CLKCON5 register to disable clock gates.

PLL Programming

Figure 5 shows the abstract PLL structure for both SPLL and UPLL. It has the multiplier coefficient, N, and divider coefficient, M, to decide the output clock ratio of N/M. There is an optional DIV2 built in the PLL to either divide down the PLL output clock by 2 or directly output it.

The PLL must always switch away from ROOT_CLK when changing any coefficients or clock sources.

When the reference clock for the phase frequency detector (PFD) has a 4 MHz input, the PLL has its best phase margin. Therefore, it is recommended that for a 16 MHz crystal clock input, configure M as 4, and for an 8 MHz crystal clock input, configure M as 2.

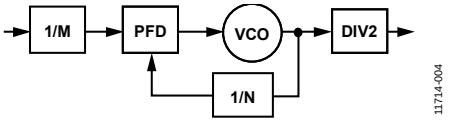


Figure 5. PLL Diagram

Table 8 shows the example and recommended settings for SPLL to output a 16 MHz or a 32 MHz clock and for UPLL to output a 60 MHz clock.

Table 8. PLL Recommended Setting

PLL	Input Clock (MHz)	M	N	DIV2	Output Clock (MHz)
SPLL	16	4	8	0	32
SPLL	16	4	16	1	32
SPLL	8	2	8	0	32
SPLL	8	2	16	1	32
SPLL	8	2	4	0	16
SPLL	8	2	8	1	16
UPLL	16	4	15	0	60
UPLL	8	2	15	0	60

The PLL settings can be programmed using the CLKCON3 and CLKCON4 registers. Both of these registers contain the same control bits for the two separate PLLs (SPLL and UPLL). To enable SPLL or UPLL, the SPLLEN or UPLLEN bit must be set, respectively. The SPLL or UPLL N multiplier can be set using the SPLLNSEL or UPLLNSEL bits, respectively. The M divider can also be set using the SPLLMSEL or UPLLMSEL bits according to the following:

- 00: divide by 1.
- 01: divide by 2.
- 10: divide by 4.
- 11: reserved.

An optional division by 2 of the input SPLL and UPLL clock can be programmed by setting the SPLLDIV2 and UPLLDIV2 bits, respectively.

Interrupts

Either PLL can interrupt the core when it locks or when it loses its lock. To enable the SPLL and UPLL interrupts, the SPLLIE and UPLLIE bits in the CLKCON3 and CLKCON4 registers must be set. The SPLLUNLOCK and UPLLUNLOCK bits in the CLKSTAT0 register indicate that the SPLL and UPLL unlock events have happened. The SPLLLOCK and UPLLLOCK bits in the CLKSTAT0 register indicate that a SPLL or UPLL lock event has happened. Both of these bits are used to interrupt the core when PLL interrupts are enabled. Both bits are sticky and must write a 1 to be cleared. These bits are different from the SPLLSTATUS and UPLLSTATUS bits in CLKSTAT0, which simply mirrors the value of the SPLL or UPLL lock signal (1 = locked, 0 = unlocked).

Regarding the SPLL or UPLL lock event, while it is fine to use the lock detect status bits to determine when the PLL clock is safe to use and locked initially, it is not advised to use the unlock sticky bits as a means to declare the PLL clocks are unsuitable; that is, it is not a reliable indicator that the PLL has lost lock. The reason for this is that the SPLL and UPLL sensitivity threshold is too low and causes the detect function to be too sensitive for conditions related to environment, application, or device.

Sequence

To start using the system PLL, the following sequence of events is recommended. The example multiplies a 16 MHz crystal input to 32 MHz via the SPLL and uses it as the root clock. The HCLK is set to 16 MHz and the PCLK to 4 MHz.

1. Enable PLL interrupts by setting SPLLIE in CLKCON3 to 0x1.
2. Set the PLL input to the external high frequency (HF) crystal (XTAL oscillator). Set PLLMUX in CLKCON0 to 0x1.
3. Enable the external crystal (HFXTAL oscillator). Set HFXTALEN in OSCCTRL to 0x1.
4. Set the clock dividers consistent with the intended system clock rates. For example, assuming a PLL output of 32 MHz sets the HCLK to 16 MHz and the PCLK to 4 MHz, set PCLKDIVCNT in CLKCON1 to 0x08 and HCLKDIVCNT in CLKCON1 to 0x02.
5. Set up the PLL M and N values and enable the PLL. Set SPLLEN in CLKCON3 to 0x1, SPLLMSEL in CLKCON3 to 0x2, and SPLLNSEL in CLKCON3 to 0x08.
6. Wait for the PLL interrupt indicating that the PLL has locked. Optionally, also check that the crystal is stable at this stage, even though the PLL must not lock if the crystal is not stable.
7. Clear the PLL interrupt and select PLL as the system clock source. Set SPLLLOCK in CLKSTAT0 to 0x1 and CLKMUX in CLKCON0 to 0x2.

The USB PLL is programmed following a simpler but similar sequence of events. Step 2 and Step 7 are not required because there is only one input to the UPLL (the HF crystal), and it is the only possible source to the USB PHY clock.

For Step 4, the UPLL requires precharging of the loop to prevent the VCO from turning off. This is a two-part process as follows:

1. Set divider ratio to precharge loop for a duration of 200 µs. Set the UPLLMSEL value = 0x2 and the UPLLNSEL value = 0x3F.
2. Set divider ratio to desired UPLL frequency. Set the UPLLMSEL value = 0x2 and the UPLLNSEL value = 0x0F.

Crystal Programming

The crystals are disabled by default and can be programmed using the OSCCTRL register. The crystals can be enabled by setting the HFXTALEN or LFXTALEN bits. The stable signal status bits are also mirrored in this register.

Note the following:

- Before issuing a SYSRESETREQ, allowing the Cortex-M3 to assert a reset request signal to the system reset generator, the HFOSCEN (OSCCTRL bit 1) must be set to ensure that all system components reset properly. This is independent of the root clock mux and SPLL clock mux settings.
- When disabling the external LFXTAL, the LFXTAKEN bit must be written to twice with a delay of at least two low frequency clock periods between write commands.

Interrupts

Each crystal can interrupt the core when its output clock becomes stable. The interrupts are enabled by setting the HFXTALIE and LFXTALIE bits in the CLKCON0 register. Register CLKSTAT0 contains the stable information pertaining to both crystals. The HFXTALSTATUS and LFXTALSTATUS bits in the CLKSTAT0 register contain the current state of the stable signals of the crystals. The HFXTALOK or LFXTALOK bits in the CLKSTAT0 register are set when an event is detected on the stable signals of the crystals. The HFXTALOK/HFXTALNOK and LFXTALOK/LFXTALNOK bits are sticky and must be cleared by writing a 1 to them. The HFXTALNOK and LFXTALNOK bits are not continuous XTAL monitors and are only set as a confirmation that the corresponding XTAL has been properly disabled.

PLL Clock Protection

In the event that the clock source to the PLL is lost, the PLL maintains operation at a reduced output frequency. This behavior allows PLL interrupt sources such as SPLLUNLOCK and UPLLUNLOCK (CLKSTAT0 Bit 6 and Bit 2) to be serviced and enables the appropriate action to be taken by the core. This feature protects the core from an indefinite stall due to broken or shorted leads of the XTAL circuit.

Oscillator Programming

Both the internal oscillators are enabled by default. Before issuing a SYSRESETREQ, allowing the Cortex-M3 to assert a reset request signal to the system reset generator, the HFOSCEN (OSCCTRL Bit 1) must be set. This ensures that all system components are reset properly. This is independent of the root clock mux and SPLL clock mux settings.

Enabling the HFOSC is done by using the HFOSCEN bit in the OSCCTRL register. Note that before stopping the HFOSC internal oscillator, the HFXTAL must be running the system. Otherwise, the device locks because its clock has been stopped by the user without possibility of recovery. Similarly, the LFOSC internal oscillator can be enabled with the LFOSCEN bit in the OSCCTRL register. Peripherals driven with a 32 kHz clock must be switched over to the LFXTAL external crystal oscillator before the LFOSC internal oscillator is disabled.

INTERNAL RC AND EXTERNAL XTAL OSCILLATORS

Platform HF RC Oscillator

The 16 MHz high frequency oscillator is enabled by the HFOSCEN bit. The RC oscillator is calibrated to ±5%.

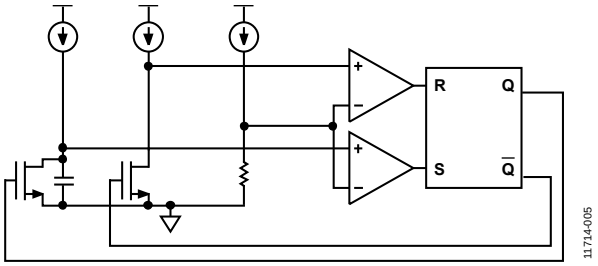


Figure 6. HF RC Oscillator Block Diagram

Platform LF RC Oscillator

The 32.768 kHz low frequency RC oscillator is enabled by the LFOSCEN bit. The RC oscillator is calibrated to ±20% and is in the 3 V domain.

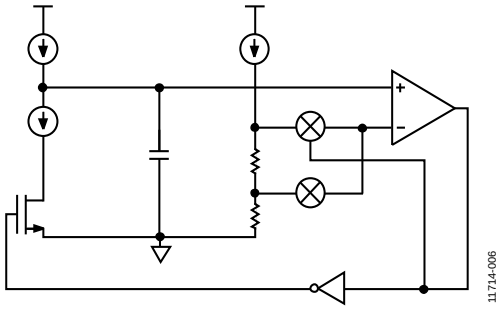


Figure 7. LF RC Oscillator Block Diagram

Platform HF XTAL Oscillator

The HF XTAL oscillator is the accurate clock source of the system, with a frequency of 16 MHz. It is used as an input for the system/USB PLL or as a direct clock source for the respective digital clock requirements and is in the 1.8 V domain.

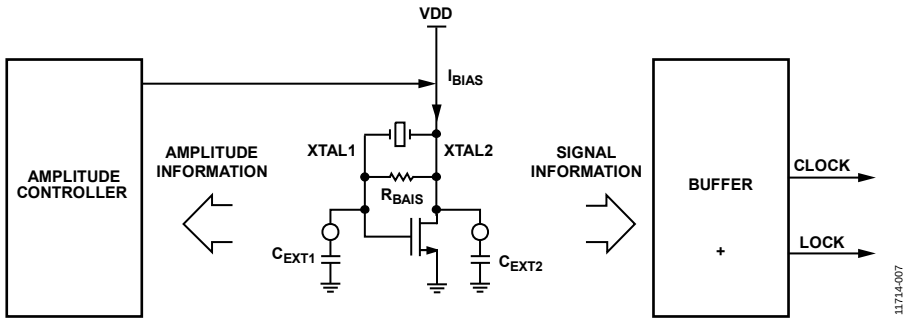


Figure 8. HF XTAL Block Diagram

Platform LF XTAL Oscillator

The LF XTAL oscillator is the clock source for the RTC. It is used to keep the time of the system. It provides a 32.768 kHz output clock with an external load of 15 pF.

Once the oscillator is enabled, it remains always on, even in hibernate and supercapacitor mode.

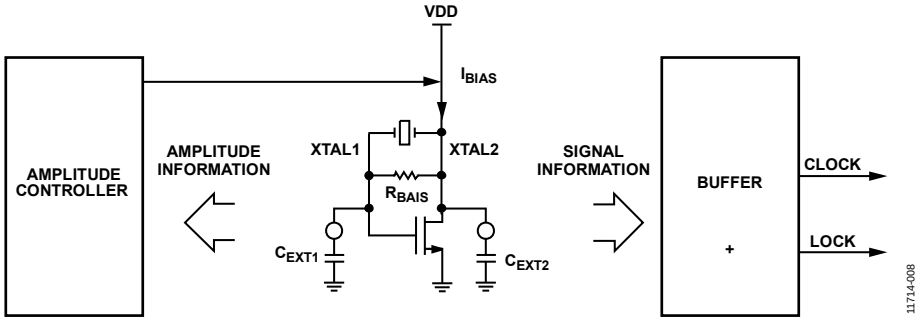


Figure 9. LF XTAL Block Diagram

EXAMPLE USE CASES

The following flow diagrams highlight the sequence of events required to set the system/USB clocks.

Set System Clock to PLL Input Source

The following three timing diagrams show the sequence of events to change the system clock from being based on an internal RC oscillator to being based on a PLL input source.

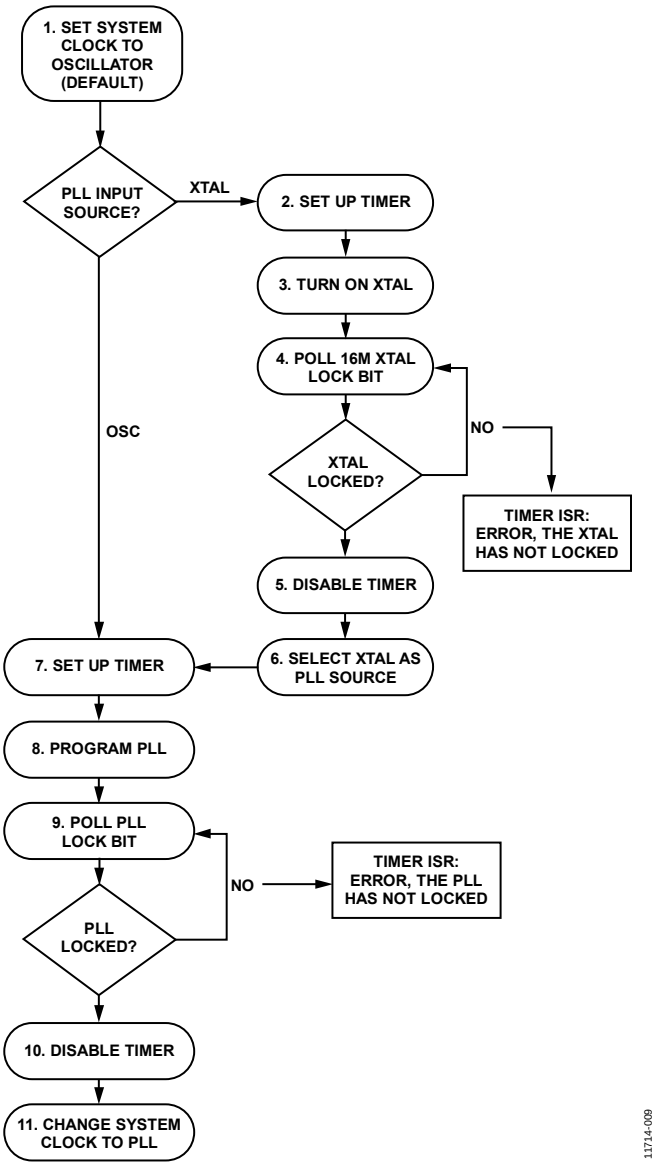


Figure 10. Change System Clock to PLL (Poll Method)

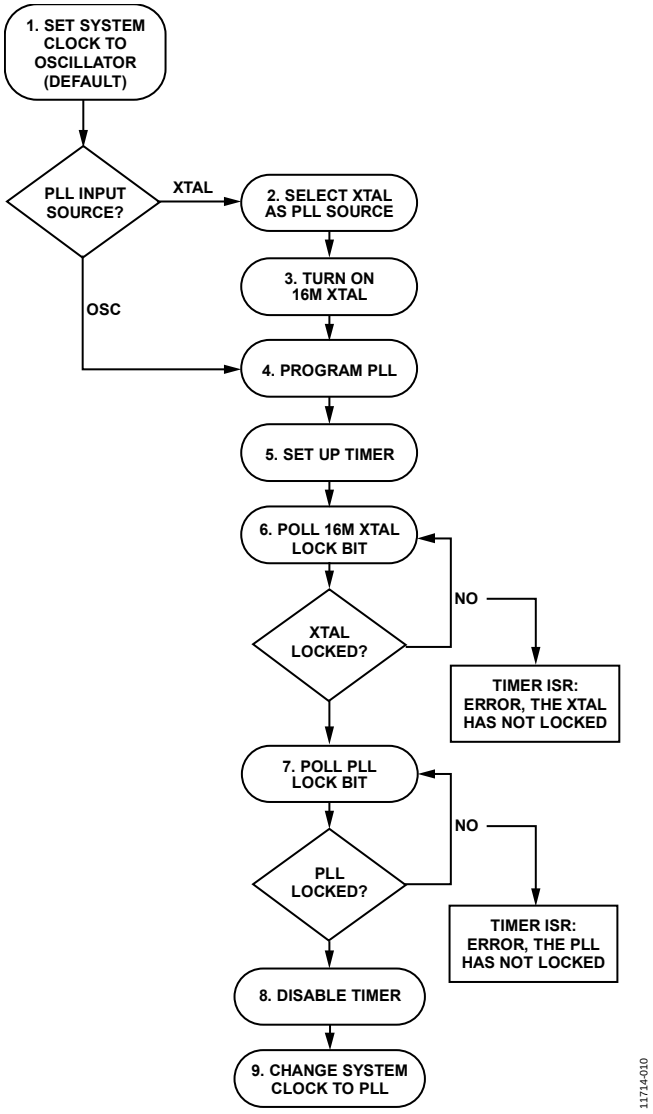


Figure 11. Change System Clock to PLL (Poll Alternative Method)

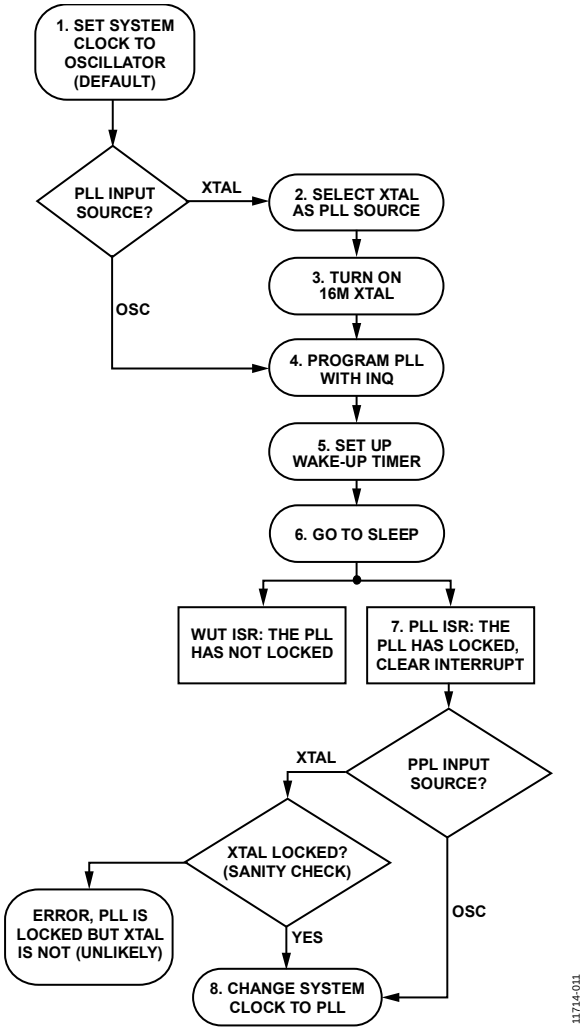


Figure 12. Change System Clock to PLL (IRQ Method)

Set System Clock to XTAL

The following two timing diagrams show the sequence of events to change the system clock from being based on an internal RC oscillator to being based on an XTAL source.

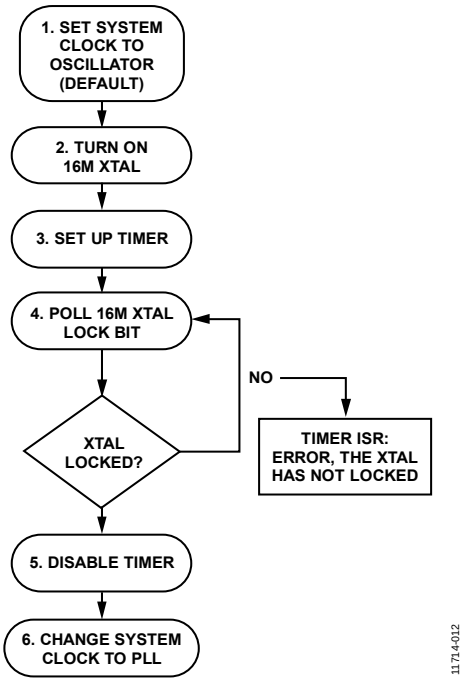


Figure 13. Change System Clock to XTAL (Poll Method)

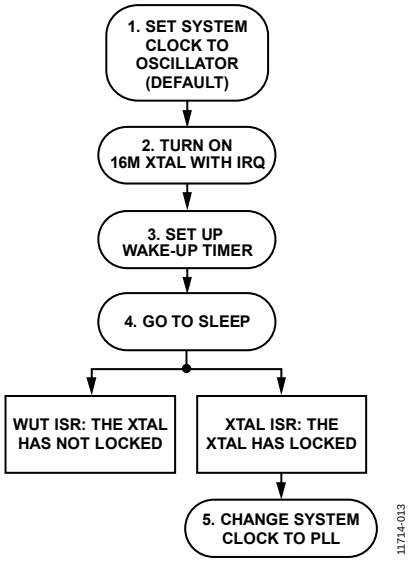


Figure 14. Change System Clock to XTAL (IRQ Method)

Changing System Clock Source

The following diagram shows the sequence to change the system clock source from being based on an RC oscillator to being based on one of the following: an XTAL, a GPIO input, or a PLL input source.

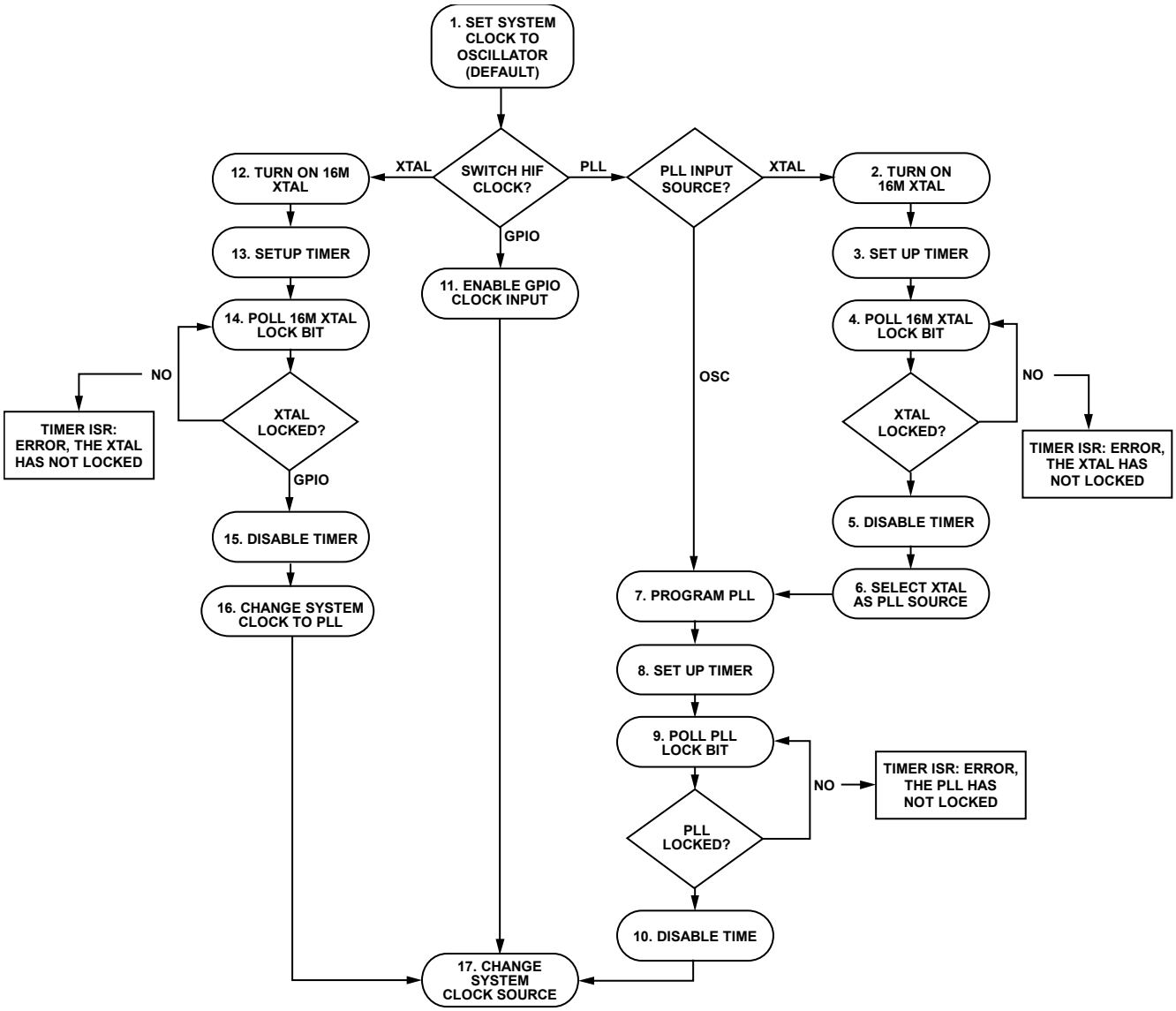


Figure 15. Changing System Clock

Clocking Control for USB

The USB controller requires a 32 MHz controller clock and a 60 MHz PHY clock, while the rest of the system can run with a 16 MHz clock. This section describes when to change the clocks.

The control of USB clocks relates to clock dividers (CLKCON1), HF XTAL and PLLs (OSCKEY, OSCCTRL, CLKCON3, CLKCON4, CLKSTAT0), and system clock multiplexer (CLKCON0).

Whenever the USB connection or resume activity is detected, switch the system clock to a 32 MHz clock and enable the UPLL. When USB is disconnected or suspended, the system clock can be switched back to 16 MHz and both the SPLL and UPLL can be disabled. For LPM sleep state, due to its fast (50 µs to 1.2 ms) wake-up requirement, the HF XTAL must remain active but UPLL can be turned off (UPLL needs about 35 µs to restart, whereas HF XTAL needs ~10 ms to restart).

Figure 16, Figure 17, and Figure 18 describe how to enable and disable the USB clocks. The USB clocks must be enabled when the USB is in use and disabled when the USB is not in use.

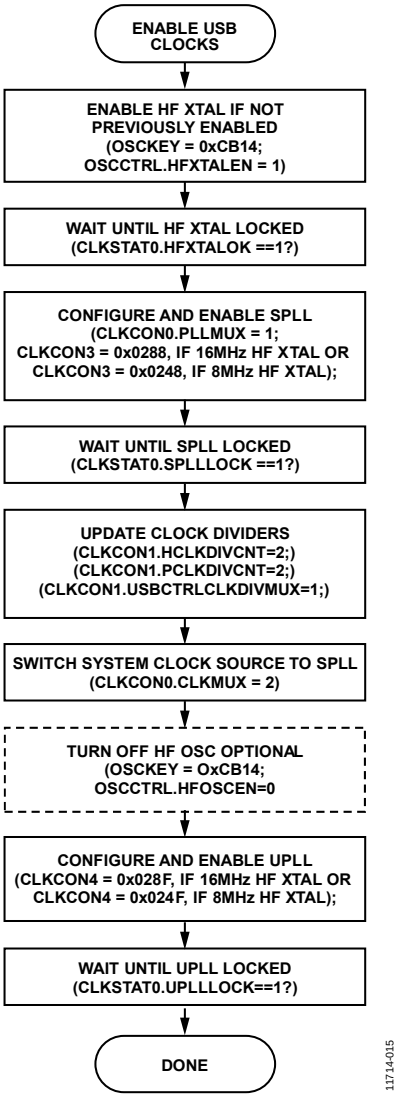


Figure 16. Enable USB Clocks

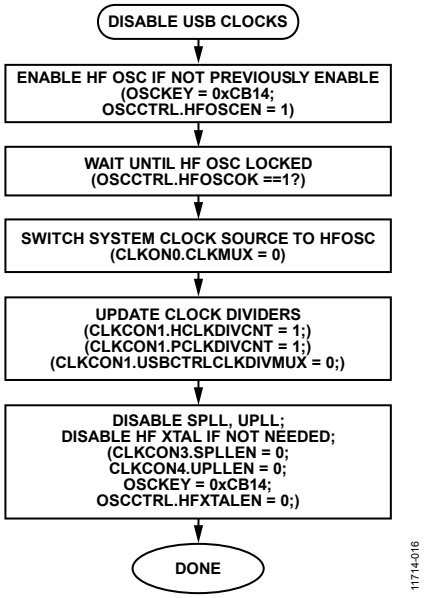


Figure 17. Disable USB Clocks

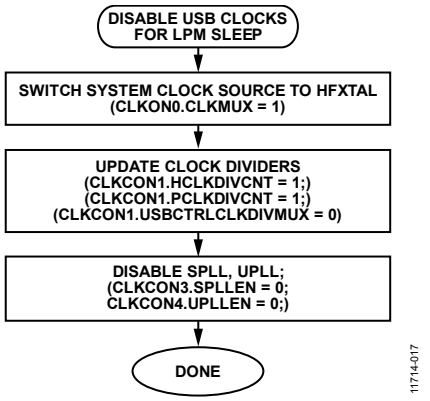


Figure 18. Disable USB Clocks For LPM Sleep

The following diagram describes how the USB transits between its working states: L0 (on), L1 (sleep), L2 (suspend), and L3 (off).

To detect the activity of the USB when transiting between these states, the USBWKUP logic is required. The USBWKUP logic has level or edge detection of USBVBUS, DP, and DM lines as controlled in EI2CFG and generates an interrupt at Position Number 36 (see Table 114) if enabled by EI2CFG.USBVBUSEN, EI2CFG.USBDMEN, and/or EI2CFG.USBDPEN. The detection events are saved at Register USBWKSTAT.

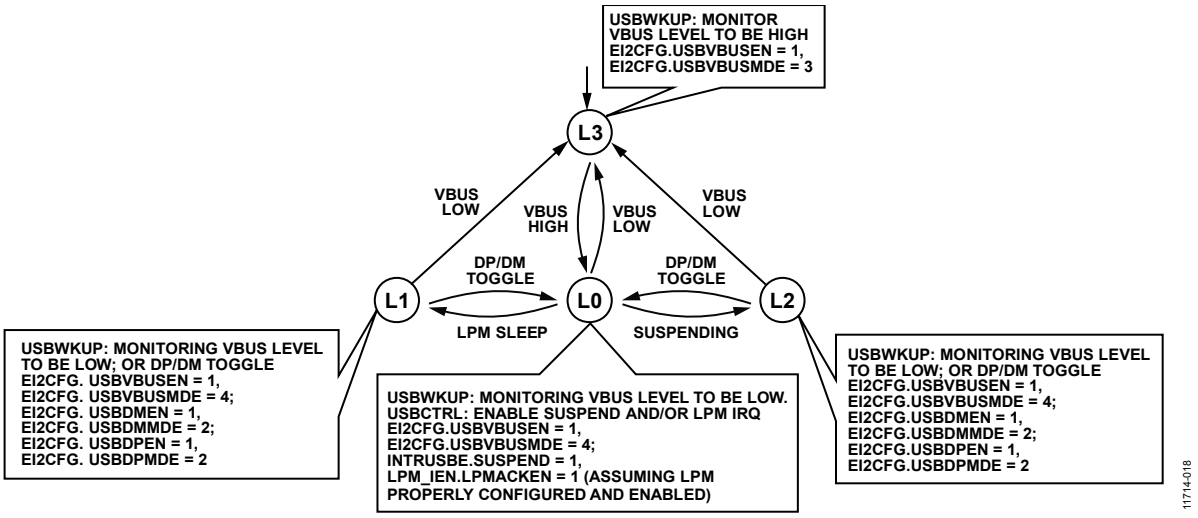


Figure 19. USB State Diagram

L3

Initially, the USB controller enters the L3 state by default and does not require the USBCTLCLK or USBPHYCLK. It can only transit to the L0 state if the USB controller is enabled and connected. To detect the USB connection while in the L3 state, configure the USBWKUP block to monitor the USBVBUS line for a high state. When USBVBUS is detected as high, call the enable USB clocks task to have the 32 MHz USBCTLCLK and the 60 MHz USBPHYCLK so that the USB can transit from L3 to L0.

L0

The active USB state requires a 32 MHz USBCTLCLK and a 60 MHz USBPHYCLK to work properly for USB transactions. When in the L0 state, the USB controller can transit to

- The L2 state if the USB data bus (DP and DM) idles for more than 3 ms.
- The L1 state if the LPM command is acknowledged.
- The L3 state if a disconnection occurs.

Enable the USBWKUP logic to monitor the VBUS level so that it can generate an interrupt when VBUS is low, which occurs when the USB is disconnected. In such a case, the disable USB clocks task can be called to use 16 MHz for rest logic to save power.

In addition, the USB controller must enable the suspend interrupt so that it can notify the user when the USB data bus idles for more than 3 ms. If this occurs, the disable USB clocks task can be called to use 16 MHz for rest logic to save power.

If LPM is supported and properly configured, enable the LPM interrupt so that the LPM command can be acknowledged. If this occurs, the disable USB clocks for LPM task can be called to use the 16 MHz HF XTAL for rest logic to save power. (The user may have to leave the PLL running if a very short resume period is required.)

L2

The suspend state does not need any USBCTLCLK or USBPHYCLK clock. The L2 state can transit to L3 when a disconnection occurs or to L0 if USB bus activity is detected.

To detect a disconnection, configure the USBWKUP logic to monitor when the VBUS line becomes low. If this occurs, the disable USB clocks task can be called to use 16 MHz for rest system if the clocks are not disabled yet. (Although the internal logic of the USB controller might appear to remain in suspend mode, this does not adversely affect the operation of the device. The internal logic of the USB controller is updated after a new connection is detected.)

To detect resume or reset activity, configure the USBWKUP logic to monitor DP (MD is redundant and might not have to be enabled). If this occurs, call the enable USB clocks task to have the 32 MHz USBCTLCLK and the 60 MHz USBPHYCLK so that the USB can transit to the L0 state.

L1

The LPM sleep state is defined for fast sleep (instead of 3 ms idling) and fast resume in the USB 2.0 standard. This state does not require low power as a compromise. The HF XTAL must remain active; otherwise, it cannot resume operation quickly enough. Optionally, UPLL can be disabled if the restart period is adequate. Otherwise, the L1 state is similar to the L2 state; therefore, the USB controller can transit to the L3 state when a disconnection occurs or to the L0 state if USB bus activity is detected. The USBWKUP logic must be configured the same as it is for the L2 state; the response is also the same as it is for the L2 state. If the UPLL is disabled, the user must enable the system clock to 32 MHz before reenabling the UPLL to ensure that pulses from the USBPHYCLK domain to the USBCTLCLK domain can be detected.

Table 9 lists the mentioned interrupts and corresponding interrupt service routine (ISR).

Table 9. Interrupts for USB Clock Control

USB Event	USB Wake-Up Monitoring or Interrupt	Service for USB Wake-Up Interrupt
New Connection (L3 to L0)	USBWKUP logic detecting VBUSVAILD for high (EI2CFG[15:12] = 0xB)	Enable USB clocks
Disconnection (L0, L1, or L2 to L3)	USBWKUP logic detecting VBUSVAILD for low (EI2CFG[15:12] = 0xC)	Disable USB clocks or disable USB clocks for LPM sleep
Suspend (L0 to L2)	INTRUSB:SUSPEND in USB controller	Disable USB clocks
LPM Sleeping (L0 to L1)	LPM_IRQ:LPMACK in USB controller	Disable USB clocks for LPM sleep
USB Resume (L1 or L2 to L0)	USBWKUP logic detecting DP toggle (EI2CFG[7:4] = 0xA)	Enable USB clocks
USB Reset (When L1 or L2; No Clocking Action When L0)	USBWKUP logic detecting DP toggle (EI2CFG[7:4] = 0xA)	Enable USB clocks

SYSTEM CLOCKS MEMORY MAPPED REGISTERS

System Clocks Register Map

Table 10. Clocking Register Summary

Address	Name	Description	Reset	RW
0x4000240C	OSCKEY	Key protection for OSCCTRL	0x0000	RW
0x40002410	OSCCTRL	Oscillator control	0x0003	RW
0x40028000	CLKCON0	Miscellaneous clock settings	0x0000	RW
0x40028004	CLKCON1	Clock dividers	0x1010	RW
0x40028008	RESERVED	Reserved	0x0000	R
0x4002800C	CLKCON3	System PLL	0x0088	RW
0x40028010	CLKCON4	USB PLL	0x008F	RW
0x40028014	CLKCON5	User clock gating control	0x0180	RW
0x40028018	CLKSTAT0	Clocking status	0x0000	RW
0x4002801C	RESERVED	Reserved	Unknown	R
0x40028810	RESERVED	Reserved	0x0000	R
0x40028814	RESERVED	Reserved	0x0000	R
0x40028818	RESERVED	Reserved	Unknown	R
0x4002881C	RESERVED	Reserved	Unknown	R
0x40028820	RESERVED	Reserved	Unknown	R
0x40028824	RESERVED	Reserved	0x0000	R
0x40028828	RESERVED	Reserved	0x0000	R
0x4002882C	RESERVED	Reserved	Unknown	R
0x4002883C	RESERVED	Reserved	0x0000	R

Key Protection for OSCCTRL Register

Address: 0x4000240C, Reset: 0x0000, Name: OSCKEY

Table 11. Bit Descriptions for OSCKEY

Bits	Bit Name	Description	Reset	Access
[15:0]	OSCKEY	The OSCCTRL register is key protected. To unlock this protection, write 0xCB14 to OSCKEY before writing to OSCCTRL. A write to any other register on the APB before writing to OSCCTRL returns the protection to the lock state.	0x0	RW

Oscillator Control Register

Address: 0x40002410, Reset: 0x0003, Name: OSCCTRL

The OSCCTRL register is key protected. To unlock this protection, write 0xCB14 to OSCKEY before writing to OSCCTRL. A write to any other register on the APB before writing to OSCCTRL returns the protection to the lock state.

Table 12. Bit Descriptions for OSCCTRL

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
11	HFXTALOK	Status of HFXTAL oscillator. This bit indicates when the crystal is stable after it is enabled. This bit is not a monitor and does not indicate a subsequent loss of stability. 0: oscillator is not yet stable or is disabled. 1: oscillator is enabled and is stable and ready for use.	0x0	R
10	LFXTALOK	Status of LFXTAL oscillator. This bit indicates when the crystal is stable after it is enabled. This bit is not a monitor and does not indicate a subsequent loss of stability. 0: oscillator is not yet stable or is disabled. 1: oscillator is enabled and is stable and ready for use.	0x0	R
9	HFOSCOK	Status of HFOSC oscillator. This bit indicates when the oscillator is stable after it is enabled. This bit is not a monitor and does not indicate a subsequent loss of stability. 0: oscillator is not yet stable or is disabled. 1: oscillator is enabled and is stable and ready for use.	0x0	R

Bits	Bit Name	Description	Reset	Access
8	LFOSCOK	Status of LFOSC oscillator. This bit indicates when the oscillator is stable after it is enabled. This bit is not a monitor and does not indicate a subsequent loss of stability. 0: oscillator is not yet stable or is disabled. 1: oscillator is enabled and is stable and ready for use.	0x0	R
[7:4]	RESERVED	Reserved.	0x0	R
3	HFXTALEN	High frequency crystal oscillator enable. This bit is used to enable/disable the oscillator. The oscillator must be stable before use. Notes that this bit is key protected by OSCKEY. 0: the HFXTAL oscillator is disabled and placed in a low power state. 1: the HFXTAL oscillator is enabled.	0x0	RW
2	LFXTALEN	Low frequency crystal oscillator enable. This bit is used to enable/disable the oscillator. The oscillator must be stable before use. Note that this bit is key protected by OSCKEY. 0: the LFXTAL oscillator is disabled and placed in a low power state. 1: the LFXTAL oscillator is enabled.	0x0	RW
1	HFOSCEN	High frequency internal oscillator enable. This bit is used to enable/disable the oscillator. The oscillator must be stable before use. This bit must be set before the SYSRESETREQ system reset can be initiated. Notes that this bit is key protected by OSCKEY. 0: the HFOSC oscillator is disabled and placed in a low power state. 1: the HFOSC oscillator is enabled.	0x1	RW
0	LFOSCEN	Low frequency internal oscillator enable. This bit is used to enable/disable the oscillator. The oscillator must be stable before use. Note that this bit is key protected by OSCKEY. 0: the LFOSC oscillator is disabled and placed in a low power state. 1: the LFOSC oscillator is enabled.	0x1	RW

Miscellaneous Clock Settings Register

Address: 0x40028000, Reset: 0x0000, Name: CLKCON0

The CLKCON0 register is used to configure clock sources used by various systems such as the core and memories as well as USB and peripherals. All unused bits are read only returning a value of 0. Writing unused bits has no effect.

Table 13. Bit Descriptions for CLKCON0

Bits	Bit Name	Description	Reset	Access
15	HFXTALIE	High frequency crystal interrupt enable. Controls if the core is interrupted on a HFXTALOK or HFXTALNOK status, or if no interrupt is generated. Never clear this bit while a core interrupt is pending. 0: an interrupt to the core is not generated on a HFXTALOK or HFXTALNOK. 1: an interrupt to the core is generated on a HFXTALOK or HFXTALNOK.	0x0	RW
14	LFXTALIE	Low frequency crystal interrupt enable. Controls if the core is interrupted on a LFXTALOK or LFXTALNOK status, or if no interrupt is generated. Never clear this bit while a core interrupt is pending. 0: an interrupt to the core is not generated on a LFXTALOK or LFXTALNOK. 1: an interrupt to the core is generated on a LFXTALOK or LFXTALNOK.	0x0	RW
[13:12]	RESERVED	Reserved.	0x0	R
11	PLLMUX	SPLL source select mux. PLLMUX selects which source clock is fed to the SPLL (PLL_MUX_SEL). The selection must be made before the SPLL is enabled. Do not change the selection after the SPLL is enabled. 0: internal RC oscillator is selected. 1: external XTAL oscillator is selected.	0x0	RW
10	FIXMASTERTYPE	Force MasterType for debugger. See the ARM Cortex-M3 Technical Reference Manual.	0x0	RW
9	RESERVED	Reserved.	0x0	RW
8	LFCLKMUX	32 kHz clock select mux. 0: internal 32 KHz oscillator is selected. 1: external 32 KHz crystal is selected.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[7:4]	CLKCOUT	GPIO clock out select. Used to select which clock is output on the selected GPIO pin (AON_GPIO_MUX_SEL). Implemented as a 16 to 1 mux. 0000: ROOT_CLK 0001: LF_CLK 0010: CTCLK 0011: HCLK_BUS 0100: HCLK_CORE 0101: PCLK 0110: USBCTRLCLK 0111: USBPHYCLK 1000: GPT0_CLK 1001: GPT1_CLK 1010: GPT2_CLK 1011: WUT_CLK 1100: RTCCNT_CLK 1101 to 1111: not used	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	R
[1:0]	CLKMUX	Clock mux select. Determines which single shared clock source is used by the PCLK, CTCLK, and HCLK dividers. Ensure that an enabled active stable clock source is selected (AON_CORE_MUX_SEL). 00: high frequency internal oscillator is selected. 01: high frequency external oscillator is selected. 10: system PLL is selected. 11: external GPIO port is selected.	0x0	RW

Clock Dividers Register

Address: 0x40028004, Reset: 0x1010, Name: CLKCON1

The CLKCON1 register is used to set the divide rates for the USBCTLCLK, HCLK, and PCLK dividers. Bit 7 is also used to select which divider value is used for the USB control clock. This register can be written to at any time. All unused bits are read only, returning a value of 0. Writing to unused bits has no effect.

Table 14. Bit Descriptions for CLKCON1

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[13:8]	PCLKDIVCNT	PCLK divide count. Determines the PCLK rate based on the following equation: PCLK = ROOT_CLK/PCLKDIVCNT. For example, if ROOT_CLK is 16 MHz and PCLKDIVCNT = 0x4, PCLK operates at 4 MHz. The value of PCLKDIVCNT takes effect after a write access to this register and typically takes two to four PCLK cycles. This register can be read at any time and can be written to at any time. The reset divider count is 0x10. Value range is from 1 to 32. Values larger than 32 are saturated to 32. Values 0 and 1 have the same results as divide by 1.	0x10	RW
7	USBCTLCLKDIVMUX	USB control clock divider mux select. This bit controls the mux selector for the USB control clock. If HCLKDIVCNT = 1, this bit has no effect (the divider for the USB control clock is also 1). 0: USB CTL CLK = HCLK (dividers are equal). 1: USB CTL CLK = 2xHCLK (divider is half of HCLKDIVCNT).	0x0	RW
6	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[5:0]	HCLKDIVCNT	HCLK and CTCLK divide count. Determines the HCLK and CTCLK rate based on the following equation: HCLK (or CTCLK) = ROOT_CLK/HCLKDIVCNT. For example, if ROOT_CLK is 16 MHz and HCLKDIVCNT = 0x1, HCLK operates at 16 MHz. HCLK must be 16 MHz for the AFE to work correctly. The value of HCLKDIVCNT takes effect after a write access to this register and typically takes two to four PCLK cycles (not HCLK cycles). This register can be read at any time and can be written to at any time. The reset divider count is 0x10. Value range is from 1 to 32. Values larger than 32 are saturated to 32. Values 0 and 1 have the same results as divide by 1.	0x10	RW

System PLL Register

Address: 0x4002800C, Reset: 0x0088, Name: CLKCON3

The CLKCON3 register is used to control the system PLL. Write to this register only when the PLL is not selected as the clock source (ROOT_CLK). All unused bits are read only, returning a value of 0. Writing to unused bits has no effect.

Table 15. Bit Descriptions for CLKCON3

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
10	SPLLIE	System PLL interrupt enable. Controls if the core is interrupted on a PLL lock/PLL unlock, or if no interrupt generated. Never clear this bit while a core interrupt is pending. 0: an interrupt to the core is not generated on a PLL lock or PLL unlock. 1: an interrupt to the core is generated on a PLL lock or PLL unlock.	0x0	RW
9	SPLLEN	System PLL enable. Controls if the PLL is enabled or placed in its low power state. This bit is only set while the SPLL is not selected as the system clock source (CLKMUX bits of CLKCON0). 0: the PLL is disabled and is in its power down state. 1: the PLL is enabled. Initially the PLL does not run at the selected frequency. After a stabilization period, the PLL locks onto the selected frequency, at which time it can be selected as a system clock source (CLKMUX bits of CLKCON0).	0x0	RW
8	SPLLDIV2	System PLL division by 2. Controls if an optional divide by two is placed on the PLL output. This guarantees a balanced output duty cycle output at the cost of doubling the PLL frequency (power). Do not modify this bit after SPLLEN is set. This bit can be written at the same time SPLLEN is set. 0: the system PLL is not divided. Its output frequency equals that selected by the N/M ratio. 1: the system PLL is divided by two. Its output frequency equals that selected by the N/M ratio with an additional /2 divide.	0x0	RW
[7:6]	SPLLMSEL	System PLL M Divider. Sets the M value used to obtain the multiplication factor N/M of the PLL. 00: M set to 1 (divide by 1). 01: M set to 2 (divide by 2). 10: M set to 4 (divide by 4). 11: reserved.	0x2	RW
[5:0]	SPLLNSEL	System PLL N multiplier. Sets the N value used to obtain the multiplication factor N/M of the PLL. The default value is 0b001000 (multiply by 8). Minimum valid value is 2 and writing 0 or 1 forces it to be 2. Do not program the SPLL to an output clock lower than 8 MHz or higher than 32 MHz.	0x8	RW

USB PLL Register

Address: 0x40028010, Reset: 0x008F, Name: CLKCON4

The CLKCON4 register is used to control the USB PLL. Only write this register when the PLL is not selected as the clock source (USBPHYCLK). All unused bits are read only, returning a value of 0. Writing to unused bits has no effect.

Table 16. Bit Descriptions for CLKCON4

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
10	UPLLIE	USB PLL interrupt enable. Controls if the core is interrupted on a PLL lock/PLL unlock or no interrupt generated. Never clear this bit while a core interrupt is pending. 0: an interrupt to the core is not generated on a PLL lock or PLL unlock. 1: an interrupt to the core is generated on a PLL lock or PLL unlock.	0x0	RW
9	UPLLEN	USB PLL enable. Controls if the PLL is enabled or placed in its low power state. 0: the PLL is disabled and is in its power-down state. 1: the PLL is enabled. Initially the PLL does not run at the selected frequency. After a stabilization period, the PLL locks onto the selected frequency.	0x0	RW
8	UPLLDIV2	USB PLL division by 2. Controls if an optional divide by two is placed on the PLL output. This guarantees a balanced duty cycle output the cost of doubling the PLL frequency (power). Do not modify this bit after UPLLEN is set. This bit can be written at the same time UPLLEN is set. 0: the USB PLL is not divided. Its output frequency equals that selected by the N/M ratio. 1: the USB PLL is divided by two. Its output frequency equals that selected by the N/M ratio plus an additional /2 divide.	0x0	RW
[7:6]	UPLLMSEL	USB PLL M divider. Sets the M value used to obtain the multiplication factor N/M of the PLL. 00: M set to 1 (divide by 1). 01: M set to 2 (divide by 2). 10: M set to 4 (divide by 4). 11: reserved.	0x2	RW
[5:0]	UPLLNSEL	USB PLL N multiplier. Sets the N value used to obtain the multiplication factor N/M of the PLL. The minimum valid value is 2 and writing 0 or 1 forces it to be 2. Never program the UPLL to output clock other than 60 MHz.	0xF	RW

User Clock Gating Control Register

Address: 0x40028014, Reset: 0x0180, Name: CLKCON5

The CLKCON5 register is used to control the gates of the peripheral UCLKs.

Table 17. Bit Descriptions for CLKCON5

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
8	CTCLKOFF	CTCLK user control. This bit disables the CTCLK. It controls the gate on the CTCLK in Power Mode 0, Power Mode 1, and Power Mode 2. In Power Mode 3, the CTCLK is always off, and this bit has no effect. This bit must be manually cleared. 0: CTCLK is enabled. 1: CTCLK is disabled.	0x1	RW
7	ACLKOFF	ACLK user control. This bit disables the ACLK and AFE_ADC_CLK. It controls the gate on the ACLK and AFE_ADC_CLK in Power Mode 0, Power Mode 1, and Power Mode 2. In Power Mode 3, the ACLK and AFE_ADC_CLK are always off, and this bit has no effect. This bit must be manually cleared. 0: ACLK is enabled. 1: ACLK is disabled.	0x1	RW
6	RESERVED	Reserved. Always returns 0 when read.	0x0	R
5	UCLKI2SOFF	I^2S clock user control. This bit disables the I^2S UCLK. It controls the gate on the I^2S UCLK in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the I^2S UCLK is always off, and this bit has no effect. Note that this bit automatically clears if I^2S is accessed via the APB bus. 0: I^2S clock is enabled. 1: I^2S clock is disabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
4	UCLKUARTOFF	UART clock user control. This bit disables the UART UCLK. It controls the gate on the UART UCLK in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the UART UCLK is always off, and this bit has no effect. Note that this bit automatically clears if UART is accessed via the APB. 0: UART clock is enabled. 1: UART clock is disabled.	0x0	RW
3	UCLKI2COFF	I^2C clock user control. This bit disables the I^2C UCLK. It controls the gate on the I^2C UCLK in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the I^2C UCLK is always off, and this bit has no effect. Note that this bit automatically clears if I^2C is accessed via the APB. 0: I^2C clock is enabled. 1: I^2C clock is disabled.	0x0	RW
2	UCLKSPIHOFF	SPIH clock user control. This bit disables the SPIH UCLK. It controls the gate on the SPIH UCLK in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the SPIH UCLK is always off, and this bit has no effect. Note that this bit automatically clears if SPIH is accessed via the APB. 0: SPIH clock is enabled. 1: SPIH clock is disabled.	0x0	RW
1	UCLKSPI1OFF	SPI1 clock user control. This bit disables the SPI1 UCLK. It controls the gate on the SPI1 UCLK in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the SPI1 UCLK is always off, and this bit has no effect. Note that this bit automatically clears if SPI1 is accessed via the APB. 0: SPI1 clock is enabled. 1: SPI1 clock is disabled.	0x0	RW
0	UCLKSPI0OFF	SPI0 clock user control. This bit disables the SPI0 UCLK. It controls the gate on the SPI0 UCLK in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the SPI0 UCLK is always off, and this bit has no effect. Note that this bit automatically clears if SPI0 is accessed via the APB. 0: SPI0 clock is enabled. 1: SPI0 clock is disabled.	0x0	RW

Clocking Status Register

Address: 0x40028018, Reset: 0x0000, Name: CLKSTAT0

The CLKSTAT0 register is used to monitor PLL and oscillator status. With interrupts enabled, the user is free to continue to run initialization code or idle the core while clock components stabilize.

Table 18. Bit Descriptions for CLKSTAT0

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved. Always returns 0 when read.	0x0	R
14	HFXTALNOK	HF crystal not stable. This bit indicates the XTAL was successfully disabled. This bit is not associated with continuous monitoring of the XTAL and is not be set in the event the XTAL becomes unstable. This bit is sticky. Write a 1 to this location to clear it. If enabled, an interrupt can be associated with this bit. 0: HF crystal stable signal has not been deasserted. 1: HF crystal stable signal has been deasserted.	0x0	RW1C
13	HFXTALOK	HF crystal stable. This bit is sticky. It is used to interrupt the core when interrupts are enabled. Write a 1 to this location to clear it. 0: HF crystal stable signal has not been asserted. 1: HF crystal stable signal has been asserted.	0x0	RW1C
12	HFXTALSTATUS	HF crystal status. This bit assists in determining when the XTAL is initially stable and ready to use. This bit does not perform a continuous monitoring function and does not clear in the event an XTAL becomes unstable. 0: HF crystal is not stable or not enabled. 1: HF crystal is stable.	0x0	R
11	RESERVED	Reserved. Always returns 0 when read.	0x0	R

Bits	Bit Name	Description	Reset	Access
10	LFXTALNOK	LF crystal not stable. This bit indicates the XTAL was successfully disabled. This bit is not associated with continuous monitoring of the XTAL and is not set in the event the XTAL becomes unstable. This bit is sticky. Write a 1 to this location to clear it. If enabled, an interrupt can be associated with this bit. 0: LF crystal stable signal has not been deasserted. 1: LF crystal stable signal has been deasserted.	0x0	RW1C
9	LFXTALOK	LF crystal stable. This bit is sticky. It is used to interrupt the core when interrupts are enabled. Write a 1 to this location to clear it. 0: LF crystal stable signal has not been asserted. 1: LF crystal stable signal has been asserted.	0x0	RW1C
8	LFXTALSTATUS	LF crystal status. This bit assists in determining when the XTAL is initially stable and ready to use. This bit does not perform a continuous monitoring function and does not clear in the event an XTAL becomes unstable. 0: LF crystal is not stable or not enabled. 1: LF crystal is stable.	0x0	R
7	RESERVED	Reserved. Always returns 0 when read.	0x0	R
6	UPLLUNLOCK	USB PLL unlock. This bit is sticky. UPLLUNLOCK is set when the PLL loses its lock. UPLLUNLOCK is used as the interrupt source to signal the core that a lock was lost. Writing a 1 to this bit clears it. UPLLUNLOCK does not set again unless the USB PLL gains a lock and subsequently loses it again. 0: no loss of PLL lock was detected. 1: a PLL loss of lock was detected.	0x0	RW1C
5	UPLLLOCK	USB PLL lock. This bit is sticky. UPLLLOCK is set when the PLL locks. UPLLLOCK is used as the interrupt source to signal the core that a lock was detected. Writing a 1 to this bit clears it. UPLLLOCK does not set again unless the USB PLL loses lock and subsequently locks again. 0: no PLL lock event was detected. 1: a PLL lock event was detected.	0x0	RW1C
4	UPLLSTATUS	USB PLL status. Indicates the current status of the PLL. Initially the USB PLL is unlocked. After a stabilization period the PLL locks and is ready for use as the USB clock source. This is a read only bit. A write has no effect. 0: the PLL is not locked or not properly configured. The PLL is not ready for use as the USB clock source. 1: the PLL is locked and is ready for use as the USB clock source.	0x0	R
3	RESERVED	Reserved. Always returns 0 when read.	0x0	R
2	SPLLUNLOCK	System PLL unlock. This bit is sticky. SPLLUNLOCK is set when the PLL loses lock. SPLLUNLOCK is used as the interrupt source to signal the core that a lock was lost. Writing a 1 to this bit clears it. SPLLUNLOCK does not set again unless the system PLL gains a lock and subsequently loses it again. 0: no loss of PLL lock was detected. 1: a PLL loss of lock was detected.	0x0	RW1C
1	SPLLLOCK	System PLL lock. This bit is sticky. SPLLLOCK is set when the PLL locks. SPLLLOCK is used as the interrupt source to signal the core that a lock was detected. Writing a 1 to this bit clears it. SPLLLOCK does not set again unless the system PLL loses lock and subsequently locks again. 0: no PLL lock event was detected. 1: a PLL lock event was detected.	0x0	RW1C
0	SPLLSTATUS	System PLL status. Indicates the current status of the PLL. Initially, the system PLL is unlocked. After a stabilization period, the PLL locks and is ready for use as the system clock source. This is a read only bit. A write has no effect. 0: the PLL is not locked or not properly configured. The PLL is not ready for use as the system clock source. 1: the PLL is locked and is ready for use as the system clock source.	0x0	R

REAL-TIME CLOCK

SUMMARY

The ADuCM350 has a real-time clock (RTC) that keeps track of wall time using an externally attached 32,768 Hz crystal to generate a 1 Hz time base. The RTC maintains a count in seconds of elapsed time from a fixed reference point. This time zero reference is defined (implied) by the instantaneous elapsed value that is programmed into the RTC counter during manufacture.

The RTC operates from a dedicated voltage domain, VBACK, which is always powered, even when the ADuCM350 core is disabled. The VBACK domain is normally fed from VCCM but is backed up by a trickle charged supercapacitor. Use of the backup is intended to facilitate changing the coin battery of the ADuCM350 while ensuring that the RTC remains counting.

The RTC also has an alarm feature that interrupts the processor of the ADuCM350 when a programmed alarm value matches the RTC count.

Software is responsible for enabling and configuring the RTC and for interpreting its count value to turn this into the time of day. The RTC logic also has a digital trim capability that is calibrated at manufacturing to achieve higher ppm accuracy in tracking time.

FEATURES

The RTC meets the following performance criteria:

- Contains a low power crystal oscillation circuit that operates in conjunction with a 32,768 Hz external crystal.
- Achieves 25 ppm performance in keeping time at 25°C when used with a 10 ppm crystal and C0G class load capacitors.
- Runs for 12 hours at 25°C from a fully charged 0.08 F supercapacitor.

The RTC logic has the following features:

- A 32-bit count register of the time in seconds from a known reference point. This register is programmed under software control.
- A prescaler that divides down the 32,768 Hz crystal input to 1 Hz that is used to advance the seconds count. Note that when programming (initializing) or reenabling the count, the prescaler is automatically zeroed so that a value in seconds is deposited in the counter on exact, coincident 1 sec, and 1 minute boundaries.
- An optionally enabled alarm mode that causes a processor interrupt when the RTC count equals the alarm value. Note that such an interrupt wakes up the processor if the latter is in a sleep state.
- An RTC fail flag (sticky error flag) that is set either by the VBACK power-on reset during power up from a nonbacked up state or if the VBACK supply to the RTC falls below a low threshold voltage trip point of 1.62 V for a period of 200 ms or more. When asserted, this fail flag indicates that the RTC time count is not reliable and that processor intervention is necessary. The RTC fail flag is one of two sources of interrupt for the CPU. (The other is the alarm.) An interrupt due to an RTC failure is always enabled.
- A digital trim capability whereby a positive or negative adjustment (in units of whole seconds) can be added to the RTC count at a fixed interval to keep the RTC ppm time accuracy within target. The values for both the adjustment and interval are calculated at manufacturing and stored in flash memory.
- The 32,768 Hz input clock to the RTC can be muxed off chip using the GPIO function of the ADuCM350 to facilitate the calculation of the trim value for the RTC. The frequency of the clock is then measured by a precision counter. The required trim is subsequently written to flash for later retrieval by the CPU when initializing the RTC.

BLOCK DIAGRAM

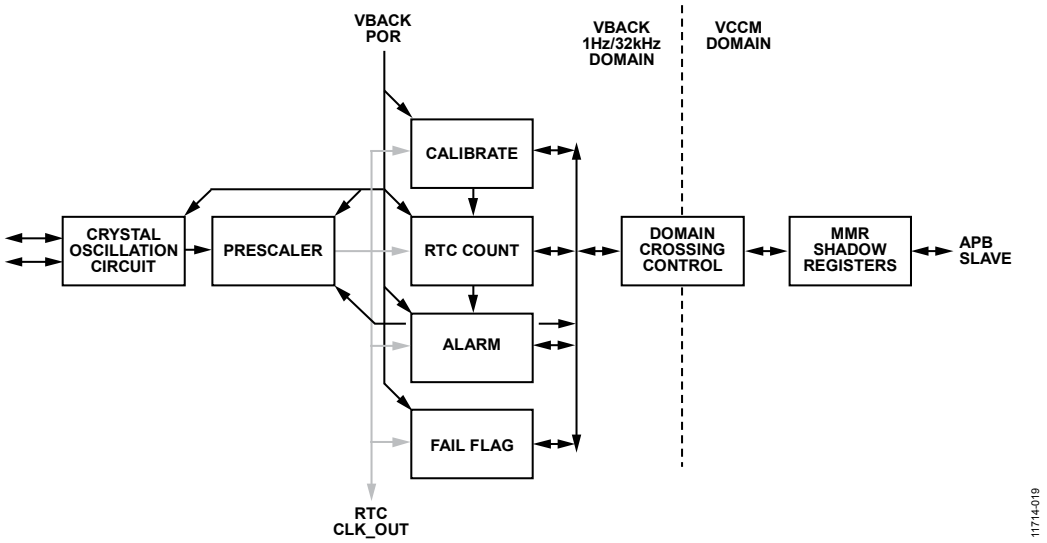


Figure 20. Real-Time Clock

OPERATION

The RTC operates in the VBACK voltage domain, which under normal conditions is continuously powered. However, when a coin battery is attached to the ADuCM350 for the first time (or if a spent battery is replaced after the supercapacitor has been depleted), a VBACK power-on reset occurs, which resets all RTC registers and also causes the RTC fail flag to be set.

An RTC reset (including resetting the fail flag) likewise occurs if the RTC is already operating but the existing VBACK supply dips below the VLO trip point of 1.62 V for 200 ms or more, indicating imminent battery depletion.

When there is sufficient battery power to enable the core voltage domain in which the CPU operates, the CPU responds to an RTC failure. If the CPU is in sleep mode, an RTC failure causes the CPU to wake up.

Note that there is no loss of any alarm event that coincides with changing the battery for the ADuCM350. The resulting interrupt due to the alarm, assuming it is enabled, is maintained by the RTC so that the CPU subsequently detects it when power is restored to the VCCM domain.

The RTC registers, which are synchronous to the 32 kHz clock in the VBACK domain, have a shadow register in the core domain. These shadow registers operate at the higher core clock rate. Any operations (reads or writes) by the CPU on the RTC use the shadow registers as staging posts. Values are synchronized from the shadow registers to the slower 32 kHz clock before the main RTC logic in the VBACK domain updates.

Note that if a write by the CPU to the RTC is pending and has not yet completed transferring across to the 1 Hz domain, a second or subsequent write by the CPU to the same RTC register cannot be stacked up or cannot overwrite the pending transaction. Any such attempts are ignored by the RTC.

To facilitate the updating of the ADuCM350 LCD display with a value based on the real-time count of the RTC, the RTC has the capability to interrupt the CPU using logic in the VCCM domain. The period of this interrupt is once every 60 sec, and its positioning relative to a 1 minute boundary can be specified by the CPU. The CPU can optionally enable or disable this interrupt.

One scenario where an RTC alarm interrupt is not issued is the following:

1. An RTC alarm value is defined in RTCALM0, RTCALM1.
2. The RTC count is subsequently redefined in the RTCCNT0 and RTCCNT1 registers to equal the just defined alarm value.
3. The RTC alarm interrupt is not issued despite the fact that the alarm interrupts are enabled.

RTC TRIMMING

The following describes a method to compute the LF crystal trim values to compensate the RTC. This can come from a static measure (a frequency counter), a real-time drift measure based on a USB transaction, Ethernet NTP, PTP protocol, or some other external reference.

Commercial crystals typically run between 20 ppm and 100 ppm. This example demonstrates trimming a particular crystal and board configuration in which an untrimmed error of about +58.6 ppm (0.00586%) was measured. This corresponds to a raw clock about 35.4 sec/week fast (30 minutes/year).

Some background calculations include the following:

- Number of seconds in a week = 7 days × 24 hours × 60 minutes × 60 seconds = 604,800 seconds.
- If ppm deviation is 58.6, then cumulative inaccuracy after a week is $58.6 \times 10^{-6} \times 604{,}800 = 35.44128$ sec (fast or slow).

Table 19 lists the RTC trim correction values for various trim intervals. Note that if there are different combinations yielding the same desired correction, the shortest trim interval (and smallest trim value) is preferred to minimize instantaneous drift.

Table 19. RTC Trim Correction (Values in ppm)

Trim Value (sec)	Trim Interval 2^{14}	Trim Interval 2^{15}	Trim Interval 2^{16}	Trim Interval 2^{17}
0	0	0	0	0
1	61.04	30.52	15.26	7.63
2	122.07	61.04	30.52	15.26
3	183.11	91.55	45.78	22.89
4	244.14	122.07	61.04	30.52
5	305.18	152.59	76.29	38.15
6	366.21	183.11	91.55	45.78
7	427.25	213.62	106.81	53.41

Referencing Table 19, the closest matching RTC trim correction for our example is 61.04 ppm. Therefore, we choose a trim interval of 2^{14} sec with a negative trim value of 1 sec, subtracting 1 sec every 4.5 hours to slow the fast crystal down to a more reasonable rate. This particular trim leaves a residual error of negative 2.44 ppm (0.000244%), making the trimmed clock a little slow (less than 1.5 sec/week, or about 1.3 minutes/year), but much better than the untrimmed accuracy of 30 minutes/year. See the RTC Trim Register section for more details.

RECOMMENDATIONS FOR USING THE RTC

Entering Hibernation or Stopping PCLK

Before entering hibernation mode or any mode that causes PCLK to stop, the CPU must first wait until there is confirmation from the RTC that no previously posted writes have yet to complete. The CPU can check this by reading the RTCSR0 register.

This step is necessary to ensure that the power gated half of the RTC does not miss a transaction handshake (during execution of a posted write), returning from the 32 kHz always on RTC domain while the power gated core is in hibernation or has its clock stopped.

Ensuring No Communication Across the RTC Power Boundary When Battery Depletion Is Imminent

When the power loss to the core of the ADuCM350 is imminent due to battery depletion, take the following course of action to ensure the integrity of the RTC:

- Cancel all writes to the RTC by writing a cancellation key of 0xA2C5 to the RTCGWY register, which takes effect immediately.
- Do not post any further register writes to the RTC until power is lost by the core.

These steps ensure that no communication occurs between the CPU and the RTC when the isolation barrier of the RTC power domain is subsequently activated.

RTC MEMORY MAPPED REGISTERS

The RTC is controlled by eight 16-bit memory-mapped registers.

RTC Register Map

Table 20. RTC Register Summary

Address	Name	Description	Reset	RW
0x40002600	RTCCR	RTC control	0x03C4	RW
0x40002604	RTCSR0	RTC Status 0	0x3F81	RW
0x40002608	RTCSR1	RTC Status 1	0x0078	R
0x4000260C	RTCCNT0	RTC Count 0	0x0000	RW
0x40002610	RTCCNT1	RTC Count 1	0x0000	RW
0x40002614	RTCALM0	RTC Alarm 0	0xFFFF	RW
0x40002618	RTCALM1	RTC Alarm 1	0xFFFF	RW
0x4000261C	RTCTRM	RTC trim	0x0018	RW
0x40002620	RTCGWY	RTC gateway	0x0000	RW
0x40002624	RESERVED	Reserved—Analog Devices factory use only	0x0000	R

RTC Control Register

Address: 0x40002600, Reset: 0x03C4, Name: RTCCR

RTCCR is the main control register for the RTC. All operations are enabled and disabled by the CPU using RTCCR.

Table 21. Bit Descriptions for RTCCR

Bits	Bit Name	Description	Reset	Access
15	WPENDINTEN	Enable WPENDINT sourced interrupts to the CPU. WPENDINTEN is an enable for RTC interrupts to the CPU based on the WPENDINT sticky interrupt source field of the RTCSR0 memory mapped register (MMR). WPENDINT is activated whenever a pending slot for an MMR frees up due to a posted write being dispatched for execution. These slots are drop points for new posted writes by the CPU. The slots are one deep and there is one slot per MMR. These slots are used whenever a posted write involves updating either a whole MMR or some of its bit fields that are sourced in the slower 32 kHz clock domain. Note the distinction between the WPEND status and the WSYNC status of an MMR. The former indicates whether the RTC can accept a new posted write by the CPU to an MMR, whereas the latter indicates whether the effects of a posted write to an MMR are subsequently visible to the CPU. See the description of the WSYNCINT and WPENDINT fields of RTCSR0 for more details. 0: disable WPENDINT sourced interrupts to the CPU. 1: enable WPENDINT sourced interrupts to the CPU.	0x0	RW
14	WSYNCINTEN	Enable WSYNCINT sourced interrupts to the CPU. WSYNCINTEN is an enable for RTC interrupts to the CPU based on the WSYNCINT sticky interrupt source field of the RTCSR0 MMR. WSYNCINT is activated whenever the effects of a posted write to a 32 kHz sourced MMR or MMR bit field become visible to the CPU. The delay between the posting and being able to see the results of the transaction is due to the queuing time behind other posted transactions plus the synchronization time between clock domains when the posted write is actually executed. Note the distinction between the WPEND status and the WSYNC status of an MMR. The former indicates whether the RTC can accept a new posted write by the CPU to an MMR, whereas the latter indicates whether the effects of a posted write to an MMR are subsequently visible to the CPU. See the description of the WSYNCINT and WPENDINT fields of RTCSR0 for more details. 0: disable WSYNCINT-sourced interrupts to the CPU. 1: enable WSYNCINT-sourced interrupts to the CPU.	0x0	RW

Bits	Bit Name	Description	Reset	Access
13	WPENDERRINTEN	Enable WPENDERRINT sourced interrupts to the CPU when an RTC register write pending error occurs. Write pending errors happen if the CPU tries to countermand a register write that is already pending but has not yet been dispatched for execution. In such circumstances, the RTC rejects the attempt of the CPU to write a new value to the register. The pending write is executed instead when its turn arrives in the transaction queue. This queue operates on a first in, first out basis. A maximum of one pending write transaction per MMR is supported by the RTC. These writes are carried out in the order in which they are received by the RTC. Register writes take time to come to the front of the queue for dispatch and then to actually execute because of the difference in frequencies and synchronization between the core clock domain and the much slower 32 kHz domain within the RTC. Write pending errors, in which the RTC rejects an attempt by the CPU to post a write when there is no room to receive it, can be avoided by the CPU by first checking the pending status of a register in RTCSR1 before undertaking a write to that register. If a WPENDERRINT error occurs, the RTC interrupts the CPU if WPENDERRINTEN in RTCCR enables this course of action. When a write pending error interrupt is activated, the CPU can inspect the WERRCODE field of the RTCSR1 status register to determine which write transaction caused the first such pending error because the WPENDERRINT interrupt source was last cleared by the processor. Note the distinction between the WPEND status and the WSYNC status of an MMR. The former indicates whether the RTC can accept a new posted write by the CPU to an MMR, whereas the latter indicates whether the effects of a posted write to an MMR are subsequently visible to the CPU. See the description of the WSYNCINT and WPENDINT fields of RTCSR0 for more details. 0: do not enable interrupts if write pending errors occur in the RTC. 1: enable interrupts for write pending errors in the RTC.	0x0	RW
12	ISOINTEN	Enable ISOINT sourced interrupts to the CPU when isolation of the RTC power domain is activated and subsequently deactivated. ISOINTEN enables interrupts to the CPU based on the ISOINT sticky interrupt source in the RTCSR0 status register. When power loss is imminent to all power domains on the device apart from the RTC, the RTC activates its isolation barrier so that it can continue to operate independently of the core. When power is subsequently restored to the rest of the device, the RTC activates the ISOINT interrupt source to act as a sticky record of the power loss event just finishing. This activation occurs as the RTC lowers its isolation barrier once it knows that the core has regained power. If enabled by ISOINTEN, the RTC interrupts the CPU based on ISOINT. The CPU can then inspect the ISOINT field of RTCSR0 to discover that the CPU has recovered from a total loss of power. 0: disable ISOINT sourced interrupts to the CPU. 1: enable ISOINT sourced interrupts to the CPU.	0x0	RW
11	LCDINTEN	Enable LCDINT sourced LCD update interrupts to the CPU. LCDINTEN allows the CPU to determine if an interrupt from the RTC to update the LCD display value occurs at the time in seconds given by LCDUPDTIM beyond a 1 minute boundary. Note that for such interrupts to occur, the detection of an LCD update event (time beyond a 1 minute boundary) must first be enabled using LCDEN. After such an event has been detected and stickily recorded, the value of LCDINTEN controls whether an interrupt is issued by the RTC for that event. 0: disable interrupts due to LCD update time. 1: enable interrupts due to LCD update time.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[10:5]	LCDUPDTIM	LCD update time in seconds beyond a 1 minute boundary. LCDUPDTIM allows the CPU to position an LCD update interrupt from the RTC at any integer number of seconds from a 1 minute boundary. Boundaries are defined in the following way. The RTC realigns itself to create coincident 1 minute and 1 sec boundaries whenever either of the following events occurs: (1) the CPU writes a new pair of values to the RTCCNT1 and RTCCNT0 registers to redefine the elapsed seconds count while the RTC is enabled; or (2) the CPU enables the RTC from a disabled state using the CNTEN field of RTCCR. Values of 0 to 59 (sixty legal values) are allowed for LCDUPDTIM. If a greater value is configured, this is treated as 0 sec. 30: example of setting an LCD update interrupt from the RTC to be issued to the CPU at 30 sec past a 1 minute boundary. 55: example of setting an LCD update interrupt from the RTC to be issued to the CPU at 55 sec past a 1 minute boundary.	0x1E	RW
4	LCDEN	Enable RTC determination of when an LCD minute display update occurs. LCDEN enables the RTC to detect and record until cleared by the CPU, the interrupt condition of RTCCNT1 and RTCCNT0 having reached an LCDUPDTIM number of seconds past a 1 minute boundary. Note that LCDEN enables the detection of this condition, whereas LCDINTEN enables the generation of a resultant interrupt. 0: disable determination of LCD update time. 1: enable determination of LCD update time.	0x0	RW
3	TRMEN	Enable RTC digital trimming. Trimming of the RTC allows the real-time count in seconds to be adjusted on a periodic basis to track time with better accuracy. TRMEN enables this adjustment, provided the RTC is enabled via CNTEN. The exact nature of the trim (period, number of seconds to be added or subtracted) is defined in the RTCTRM register. Note that if TRMEN is activated from a disabled state while CNTEN is also active, a trim interval boundary occurs and a new trim interval begins. No trim adjustment of the RTC count occurs on such TRMEN activation. A whole, enabled trim interval must have elapsed before any adjustment is made. 0: digital trimming of the RTC count value is disabled. 1: trim is enabled.	0x0	RW
2	ALMINTEN	Enable ALMINT sourced alarm interrupts to the CPU. ALMINTEN gives the CPU extra control over whether an alarm event (alarm count matches the RTC count) triggers an interrupt. Under normal conditions, ALMINTEN is set active, most notably when the detection of an alarm condition is enabled by ALMEN. Note that if ALMINTEN is active (alarm interrupts enabled) but the alarm itself is disabled (ALMEN inactive), no interrupts occur. 0: disable alarm interrupts. 1: enable an interrupt if RTC alarm and count values match.	0x1	RW
1	ALMEN	Enable the RTC alarm operation. ALMEN must be set active for the alarm logic to function and for any alarm event to be detected. Such an event is defined as a match between the values of the RTC count and alarm registers, namely RTCCNT1, RTCCNT0, RTCALM1 and RTCALM0. Count and alarm values and match conditions are all defined on a 32-bit basis, although the constituent registers are individually 16 bits wide. When enabled by ALMEN, the detection of an alarm event is held in the sticky interrupt source bit field, LCDINT, of the status register, RTCSR0. Note that for alarm detection to function and be controlled by ALMEN, the overriding CNTEN global enable for the RTC must also be active. 0: disable detection of alarm events. 1: enable detection of alarm events.	0x0	RW

Bits	Bit Name	Description	Reset	Access
0	CNTEN	Global enable for the RTC. CNTEN enables counting of elapsed real time and acts as a master enable for the RTC. Note that if the RTC is disabled via CNTEN, no time is counted, no alarm or LCD interrupt conditions are detected, no alarm or LCD interrupts are issued and no trimming occurs. However, all interrupt sources can be cleared by the CPU irrespective of the value of CNTEN. Note also that the detection and issuing of an RTCFAIL interrupt is unconditionally enabled and has no dependency on CNTEN. If the RTC is enabled by activating CNTEN, this event causes a realignment of the prescaler, the trim interval, and the modulo-60, 1 minute counter (used by the RTC to generate LCD update interrupts). The RTC initiates a 1 sec boundary, a 1 minute boundary, and a new trim interval whenever CNTEN is activated. No trim adjustment to the RTC count is made when CNTEN is activated from a disabled state. 0: disable the RTC. 1: enable the RTC.	0x0	RW

RTC Status 0 Register

Address: 0x40002604, Reset: 0x3F81, Name: RTCSR0

Information on RTC operation is made available to the CPU via the two status registers, RTCSR0 and RTCSR1. These registers include all flags related to CPU interrupt sources and error conditions within the RTC.

Note that there is a one for one correspondence in bit positions in RTCSR0 and RTCSR1 for their WSYNC<mmr> and WPEND<mmr> fields respectively. The WSYNC status for an MMR in RTCSR0 is located at the same bit position as the WPEND status for the same MMR in RTCSR1.

The distinction between WSYNC and WPEND is as follows. If a posted write transaction to an MMR in the RTC has completed execution and the effects of the transaction are visible to the processor via the APB port of the RTC, the WSYNC status for the MMR in question is set to 1. Otherwise, if the effects of a posted transaction to an MMR are not yet visible, the WSYNC status for that MMR is 0.

WPEND, on the other hand, indicates whether there is room in the RTC to accept a new posted write to a given MMR. If a previously posted write to an MMR is awaiting execution and is occupying the drop point for posted writes to that MMR, the WPEND status of the MMR concerned is 1. Otherwise, if the RTC has room to accept a new posted write for an individual MMR, the WPEND status for that MMR is 0.

Posted writes take time to complete if they concern MMRs or MMR fields that are sourced in the 32 kHz clock domain of the RTC. These MMRs (all fields) are as follows: RTCCR, RTCCNT0, RTCCNT1, RTCALM0, RTCALM1, RTCTRM. The following sticky interrupt source fields of the RTCSR0 MMR are also sourced in the 32 kHz domain: RTCFAIL, ALMINT, LCDINT, ISOINT. Any write one to clear clearances of these interrupt source fields results in a posted write transaction for the RTCSR0 MMR, which determines the WPEND status of the RTCSR0 MMR itself.

Note that posted clearances of interrupt source fields in RTCSR0 are immediately visible by the processor, as the RTC masks them while it queues up and executes the write transaction to clear them. For this reason, the WSYNC status of RTCSR0 is always 1, to confirm that results of writes are immediately available to the CPU. That being said, the RTCSR0 does have a WPEND status, which takes on a value of 1 if there is no room in the RTC to accept another posted write clearance to a field in the RTCSR0 MMR which is sourced in the 32 kHz domain.

Note that the WPEND status and WSYNC status for the RTCSR0 are encompassing values for the MMR as a whole. There is no distinction made in these status values as to how many interrupt source fields are being cleared at the same time by a single posted write to RTCSR0.

The RTCSR1 status register, in comparison, is a read only status register. RTCSR1 has no WSYNC or WPEND status values, as there are never any posted writes to the MMR.

Table 22. Bit Descriptions for RTCSR0

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	This field is reserved for Analog Devices factory use only and reads back as all zeros.	0x0	R
13	WSYNCTRM	Synchronization status of posted writes to RTCTRM. WSYNCTRM indicates if the effects of a posted write to RTCTRM are visible to the CPU. If WSYNCTRM is low, a posted write to RTCTRM is currently queued up or in the process of being executed, but the results of this transaction are not yet visible to the CPU. When WSYNCTRM goes high and thereby activates the WSYNCINT sticky interrupt source, the effects of a write to RTCTRM are then available to the processor. The delay in the visibility of results is due to (1) the queuing time behind transactions that were posted earlier, and (2) the synchronization delay between RTC clock domains during the actual execution. 0: results of a posted write are not yet visible to the CPU. 1: results of a posted write are visible to the CPU.	0x1	R
12	WSYNCALM1	Synchronization status of posted writes to RTCALM1. WSYNCALM1 indicates if the effects of a posted write to RTCALM1 are visible to the CPU. If WSYNCALM1 is low, a posted write to RTCALM1 is currently queued up or in the process of being executed, but the results of this transaction are not yet visible to the CPU. When WSYNCALM1 goes high and thereby activates the WSYNCINT sticky interrupt source, the effects of a write to RTCALM1 are then available to the processor. The delay in the visibility of results is due to (1) the queuing time behind transactions which were posted earlier, (2) any waiting for a paired write to be posted to RTCALM0 (because only paired writes are ever dispatched for execution of a redefinition of the RTC alarm), or (3) the synchronization delay between RTC clock domains during the actual execution. 0: results of a posted write are not yet visible to the CPU. 1: results of a posted write are visible to the CPU.	0x1	R
11	WSYNCALM0	Synchronization status of posted writes to RTCALM0. WSYNCALM0 indicates if the effects of a posted write to RTCALM0 are visible to the CPU. If WSYNCALM0 is low, a posted write to RTCALM0 is currently queued up or in the process of being executed, but the results of this transaction are not yet visible to the CPU. When WSYNCALM0 goes high and thereby activates the WSYNCINT sticky interrupt source, the effects of a write to RTCALM0 are then available to the processor. The delay in the visibility of results is due to (1) the queuing time behind transactions which were posted earlier, (2) any waiting for a paired write to be posted to RTCALM1 (because only paired writes are ever dispatched for execution of a redefinition of the RTC alarm), or (3) the synchronization delay between RTC clock domains during the actual execution. 0: results of a posted write are not yet visible to the CPU. 1: results of a posted write are visible to the CPU.	0x1	R
10	WSYNCCNT1	Synchronization status of posted writes to RTCCNT1. WSYNCCNT1 indicates if the effects of a posted write to RTCCNT1 are visible to the CPU. If WSYNCCNT1 is low, a posted write to RTCCNT1 is currently queued up or in the process of being executed, but the results of this transaction are not yet visible to the CPU. When WSYNCCNT1 goes high and thereby activates the WSYNCINT sticky interrupt source, the effects of a write to RTCCNT1 are then available to the processor. The delay in the visibility of results is due to (1) the queuing time behind transactions which were posted earlier, (2) any waiting for a paired write to be posted to RTCCNT0 (because only paired writes are ever dispatched for execution of an RTC count definition), or (3) the synchronization delay between RTC clock domains during the actual execution. 0: results of a posted write are not yet visible to the CPU. 1: results of a posted write are visible to the CPU.	0x1	R

Bits	Bit Name	Description	Reset	Access
9	WSYNCCNT0	Synchronization status of posted writes to RTCCNT0. WSYNCCNT0 indicates if the effects of a posted write to RTCCNT0 are visible to the CPU. If WSYNCCNT0 is low, a posted write to RTCCNT0 is currently queued up or in the process of being executed, but the results of this transaction are not yet visible to the CPU. When WSYNCCNT0 goes high and thereby activates the WSYNCINT sticky interrupt source, the effects of a write to RTCCNT0 are then available to the processor. The delay in the visibility of results is due to (1) the queuing time behind transactions which were posted earlier, (2) any waiting for a paired write to be posted to RTCCNT1 (because only paired writes are ever dispatched for execution of an RTC count redefinition), or (3) the synchronization delay between RTC clock domains during the actual execution. 0: results of a posted write are not yet visible to the CPU. 1: results of a posted write are visible to the CPU.	0x1	R
8	WSYNCSR0	When WSYNCSR0 is low, it signifies a posted clearance to a 32 kHz maintained interrupt source is currently enqueued or executing, which is analogous to saying the interrupt source concerned is currently masked off due to a posted clearance. The interrupt sources which are maintained in the 32 kHz domain are the RTCFAIL, ALMINT, LCDINT, and ISOINT fields of RTCSR0. Clearing these sources incurs latency. All other interrupt sources (WPENERRINT, WSYNCINT and WPENDINT) are maintained in the PCLK clock domain and clearances of these are instantaneous. Note that any posted clearance of a 32 kHz-sourced interrupt masks the relevant source bit until the clearance is completed. This is so that the CPU sees the interrupt as being immediately cleared, which is a requirement to avoid repeated firing of the interrupt handler in the NVIC. The WSYNCSR0 bit field in the RTCSR0 MMR of the RTC tells the CPU about whether interrupt masking is in force. The WSYNCSR0 information can be used by the CPU to avoid the following scenario: If the CPU enters system sleep without first checking whether an RTC posted interrupt clearance was still enqueued or executing, it is possible for the system sleep to freeze the masking (prevent it from ending) of interrupts as a result of stopping PCLK. This frozen masking can stop the RTC from waking the CPU from system sleep, depending on which individual interrupt source had its masking frozen. Instead, the CPU must first read the RTCSR0 MMR in the RTC before deciding to enter system sleep. When it reads RTCSR0, it finds out (from WSYNCSR0) if any 32 kHz maintained interrupt source is currently masked off and if it is therefore inadvisable to enter system sleep. The CPU can then poll RTCSR0 and wait for WSYNCSR0 to be set before safely entering system sleep, knowing that any enabled interrupts are at that stage unmasked and will wake up the CPU from sleep if appropriate. Note that a write to RTCSR0 does not activate WSYNCINT (whose purpose is to confirm when the effects of APB writes are visible to the CPU), because the effects of posted clearances of interrupt sources are always immediately visible to the CPU. The CPU does not need a WSYNCINT interrupt to tell it about the visibility in this case; therefore, it does not happen. Note that posted clearances of interrupt sources (whether in the 32 kHz or PCLK clock domains) are always accepted by the RTC, as there is always room to accommodate them. This is because clearances posted at separate times accumulate into a common (cumulative) write transaction until the RTC schedules their execution. 0: posted clearances of 32 kHz sourced interrupt sources are currently enqueued or executing. 1: no posted clearances of 32 kHz sourced interrupt sources are currently enqueued or executing.	0x1	R
7	WSYNCCR	Synchronization status of posted writes to RTCCR. WSYNCCR indicates if the effects of a posted write to RTCCR are visible to the CPU. If WSYNCCR is low, a posted write to RTCCR is currently queued up or in the process of being executed, but the results of this transaction are not yet visible to the CPU. When WSYNCCR goes high and thereby activates the WSYNCINT sticky interrupt source, the effects of a write to RTCCR are then available to the processor. The delay in the visibility of results is due to both the queuing time behind transactions that were posted earlier, along with the synchronization delay between the RTC clock domains during the actual execution. 0: results of a posted write are not yet visible by the CPU. 1: results of a posted write are visible by the CPU.	0x1	R

Bits	Bit Name	Description	Reset	Access
6	WPENDINT	Write pending interrupt. WPENDINT is a sticky interrupt source which is activated whenever room frees up for the CPU to post a new write transaction to a 32 kHz sourced MMR or MMR bit field in the RTC. In response to a WPENDINT interrupt, the CPU can read the WPEND fields in RTCSR1 to determine which MMR most recently freed up as a result of the dispatch by the RTC for execution of a previously posted write transaction. Note the distinction between the WPEND status and the WSYNC status of an MMR. The former indicates whether the RTC can accept a new posted write by the CPU to an MMR, whereas the latter indicates whether the effects of a posted write to an MMR are subsequently visible to the CPU. See the description of the WSYNCINT field of RTCSR0 for more details. If the WPEND status of an MMR in RTCSR1 is active 1, any attempt by the CPU to post a further write transaction to the same MMR is rejected by the RTC, because there is no room to accept the posting. Such a scenario results in activation of the WPENDERRINT interrupt source in RTCSR0 and details of the rejected posting being stickily recorded in the WERRCODE field of RTCSR1. Any MMR or MMR bit field that does not have a WSYNC status field in RTCSR0 or a WPEND status field in RTCSR1 is independent of the 32 kHz domain. This means that writes to such MMRs or MMR fields are executed immediately, upon receipt by the RTC. These write are not posted, nor do they pend. Examples of such immediately executed writes are clearances by the CPU of the WSYNCERRINT, WSYNCINT and WPENDINT sticky interrupt source fields within RTCSR0. Like all interrupt sources in the RTC, the WPENDINT field has a corresponding enable bit in RTCCR that determines whether it (WPENDINT) contributes to the RTC peripheral interrupt, which is sent to the CPU. To enable a WPENDINT interrupt, set WPENDINTEN to 1 in RTCCR. WPENDINT is cleared by writing 1 to its bit position in RTCSR0. 0: there has been no change in the pending status of any posted write transaction in the RTC since WPENDINT was last cleared. 1: a posted write transaction has been dispatched since WPENDINT was last cleared, thus freeing up a slot for a new posted write by the CPU to the same MMR.	0x0	RW1C
5	WSYNCINT	Write synchronization interrupt. WSYNCINT is a sticky interrupt source that is activated whenever a posted write transaction to a 32 kHz sourced MMR or MMR bit field completes and whose effects are then visible to the CPU. When an RTC interrupt is generated due to activation of WSYNCINT, the CPU can read back RTCSR0 and see which MMR(s) has/have most recently changed its/their synchronization status, visible through the WSYNCCR, WSYNCSR0, WSYNCCNT0, WSYNCCNT1, WSYNCALM0, WSYNCALM1, and WSYNCTRM bit fields. Thus, by checking the WSYNC fields of RTCSR0, the CPU can identify which posted write transaction has just completed and caused the WSYNCINT interrupt source to stick (or restick if already active). WSYNCINT is cleared by writing a 1 to its bit position in RTCSR0. Note especially that clearances of any interrupt sources in RTCSR0 take immediate effect in terms of the CPU viewing those source values. For this reason, such clearances do not reactivate WSYNCINT, because the CPU requires no further confirmation as to when it can see the sources cleared. Thus, the WSYNCSR0 bit field of RTCSR0 always reads back as 1, indicating that the effects of a write transaction to RTCSR0 are visible. This doesn't mean that the interrupt sources have actually been cleared yet, as posted clearances to 32 kHz sources have to be enqueued like any other transaction. It simply means that the CPU is getting an advance view of the effects of the posted clearance. Note that the immediate clearances mentioned previously are facilitated by the RTC temporarily masking (to inactive) the values of any interrupts sources which lie in the 32 kHz domain and which have posted clearances awaiting execution. After the RTC actually executes the clearance (in line with enqueued transactions), the masking is rescinded of the interrupt sources concerned and their current value is made available to the CPU. In summary, the CPU sees the clearance of interrupt sources immediately, but only when the clearance has completed does the CPU be informed if another, later (during the clearance synchronization) activation of the same interrupt source has occurred in the interim. Although the RTC is able to clear the CPU's view of interrupt sources immediately, the actual pending and execution of such clearances take time behind the scenes. This is true of any interrupt source which is maintained in the 32 kHz domain due to queuing and the synchronization delay. These interrupt sources are namely the RTCFAIL, ALMINT, LCDINT, and	0x0	RW1C

Bits	Bit Name	Description	Reset	Access
		ISOINT bit fields within RTCSR0. A posted clearance to any combination of these causes WPENDSR0 to activate, indicating that the RTC cannot accept any further clearances of these fields until the current clearance transaction is dispatched. The pending time is dependent on how many other unrelated posted write transactions are enqueued in the RTC. If no other transactions are ahead in the queue, the pend time of the clearances is minimal because the clearances are dispatched to the 32 kHz domain without delay, thus freeing up room in the RTC to accept new posted clearances.		
		Note that the pend status of any transaction really indicates the ability of the RTC to accept another posted write to the same MMR requiring interaction with the 32 kHz domain. The WPEND fields in RTCSR0 can be viewed as indicators of room to accept a posted write. Each MMR has one independent (from any of the other MMRs) pend slot into which a write transaction can be posted by the CPU. Any MMR or MMR bit field which does not have a WSYNC status field in RTCSR0 or a WPEND status field in RTCSR1 is independent of the 32 kHz domain. This means that writes to such MMRs or MMR fields are executed immediately, upon receipt by the RTC. These write are not posted, nor do they pend. Examples of such immediately executed writes are clearances by the CPU of the WPENDERRINT, WSYNCINT, and WPENDINT sticky interrupt source fields within RTCSR0. Like all interrupt sources in the RTC, the WSYNCINT field has a corresponding enable bit in RTCCR which determines whether it (WSYNCINT) must contribute to the RTC peripheral interrupt which is sent to the CPU. To enable a WSYNCINT interrupt, set WSYNCINTEN to 1 in RTCCR. 0: the effects of a posted write transaction to a 32 kHz-sourced MMR or MMR bit field are not yet visible by the CPU. 1: the effects of a posted write transaction to a 32 kHz-sourced MMR or MMR bit field are now visible to the CPU.		
4	WPENDERRINT	Write pending error interrupt source. WPENDERRINT is a sticky interrupt source which indicates that an error has occurred because the CPU attempted to write to an RTC register while a previous write to the same register was pending execution. A maximum of one pending write transaction per MMR is supported by the RTC when such writes are to fields or MMRs which are sourced in the 32 kHz domain. These MMRs/MMR bit fields are RTCCR, RTCCNT0, RTCCNT1, RTCALM0, RTCALM1, RTCTRM and the RTCFAIL, ALMINT, LCDINT and ISOINT fields of RTCSR0. All writes are carried out in the order in which they are received. Register writes to MMRs or MMR bit fields sourced in the 32 kHz domain take time to complete because of synchronization between the core clock domain and the much slower 32 kHz domain. Synchronization errors can be avoided by the CPU by first checking in RTCSR1 the WPEND pending status of a register before undertaking a write to that register. The WPEND status of an MMR indicates whether the RTC can accept a new posted write to that MMR. If an error occur due to the CPU posting a write when there is no room, the RTC interrupts the CPU provided WPENDERRINTEN in RTCCR enables the WPENDERRINT interrupt source in RTCSR0. Note that if multiple write pending errors (i.e. rejected posted writes) occur, WPENDERRINT sticks active at the first occurrence. WERRCODE sticks in tandem with WPENDERRINT to identify the source of the first such error. Write one to WPENDERRINT to both clear WPENDERRINT and return WERRCODE to its inert (no error) value. 0: no posted write has been rejected by the RTC since WPENDINTERR was last cleared by the CPU. 1: a posted write has been rejected by the RTC due to a previously posted write to the same MMR which is still awaiting execution.	0x0	RW1C

Bits	Bit Name	Description	Reset	Access
3	ISOINT	RTC power domain isolation interrupt source. ISOINT is a sticky interrupt source which indicates whether the RTC has had to activate its power domain isolation barrier due to a power loss in the core. When the core regains power, the CPU can read ISOINT to inform itself of such a power event. When power loss is imminent to all power domains on the device apart from the RTC, the RTC activates its isolation barrier so that it can continue to operate independently of the core. When power is subsequently restored to the rest of the device, the RTC activates the ISOINT interrupt source to act as a sticky record of the power loss event just finishing. This activation occurs as the RTC lowers its isolation barrier once it knows that the core has regained power. If enabled by ISOINTEN, the RTC interrupts the CPU based on ISOINT. The CPU can then inspect the ISOINT field of RTCSR0 to discover that it (the CPU) has recovered from a total loss of power. To enable an RTC interrupt to the CPU due to the activation of ISOINT, set ISOINTEN to 1 in RTCCR. ISOINT can be cleared by the CPU by writing 1 to its bit position in RTCSR0. 0: the always on RTC power domain has not activated its isolation from the core since the ISOINT interrupt source was last cleared by the CPU. 1: the always on RTC power domain has activated and subsequently deactivated its isolation from the core due to a power event. This event occurred since ISOINT was last cleared by the CPU.	0x0	RW1C
2	LCDINT	LCD update interrupt source. LCDINT is a sticky flag which is the source of an optionally enabled interrupt to the CPU. This interrupt, which is activated once per minute in keeping with the LCDUPDTIM setting of RTCCR, indicates that the LCD display value must be updated to reflect the RTC count. Write one to clear the LCDINT interrupt source. 0: an LCD update interrupt is not outstanding. 1: an LCD update interrupt has occurred.	0x0	RW1C
1	ALMINT	Alarm interrupt source. ALMINT is a sticky flag which is the source of an optionally enabled interrupt to the CPU. ALMINT indicates that an alarm event has occurred due to a match between the RTC count and alarm register values. The detection of such an event is enabled by ALMEN in RTCCR, assuming CNTEN is also enabled. Write one to clear the ALMINT interrupt source. 0: no alarm has occurred. 1: an alarm interrupt has occurred.	0x0	RW1C
0	RTCFAIL	RTC failure interrupt source. RTCFAIL is a sticky flag which indicates that the voltage level in the VBACK domain has either powered up from a non-backed up state or else has dipped to such an extent as to make the RTC count value unreliable. This flag is set either by the VBACK power-on reset circuit or if the VBACK supply falls below the VLOTRIP point for 200 ms or more. RTCFAIL is the source of an always-enabled interrupt to the CPU. Write one to clear this interrupt. 0: RTC count value is reliable. 1: RTC failure. RTC count value is unreliable.	0x1	RW1C

RTC Status 1 Register

Address: 0x40002608, Reset: 0x0078, Name: RTCSR1

Information on RTC operation is made available to the CPU via the two status registers, RTCSR0 and RTCSR1. These registers include all flags related to CPU interrupt sources and error conditions within the RTC.

There is a one for one correspondence in bit positions in RTCSR0 and RTCSR1 for their WSYNC<mmr> and WPEND<mmr> fields, respectively. The WSYNC status for an MMR in RTCSR0 is located at the same bit position as the WPEND status for the same MMR in RTCSR1.

The distinction between WSYNC and WPEND is as follows. If a posted write transaction to an MMR in the RTC has completed execution and the effects of the transaction are visible to the processor via the APB port of the RTC, the WSYNC status for the MMR in question is set to 1. Otherwise, if the effects of a posted transaction to an MMR are not yet visible, the WSYNC status for that MMR is 0.

WPEND, on the other hand, indicates whether there is room in the RTC to accept a new posted write to a given MMR. If a previously posted write to an MMR is awaiting execution and is occupying the drop point for posted writes to that MMR, the WPEND status of the MMR concerned is 1. Otherwise, if the RTC has room to accept a new posted write for an individual MMR, the WPEND status for that MMR is 0.

Posted writes take time to complete if they concern MMRs or MMR fields which are sourced in the 32 kHz clock domain of the RTC. These MMRs (all fields) are as follows: RTCCR, RTCCNT0, RTCCNT1, RTCALM0, RTCALM1, RTCTRM. The following sticky interrupt source fields of the RTCSR0 MMR are also sourced in the 32 kHz domain: RTCFAIL, ALMINT, LCDINT, ISOINT. Any write one to clear clearances of these interrupt source fields results in a posted write transaction for the RTCSR0 MMR, which determines the WPEND status of the RTCSR0 MMR itself.

Note that posted clearances of interrupt source fields in RTCSR0 are immediately visible by the processor, as the RTC masks them while it queues up and executes the write transaction to clear them. For this reason, the WSYNC status of RTCSR0 is always 1, to confirm that results of writes are immediately available to the CPU. That being said, the RTCSR0 does have a WPEND status which takes on a value of 1 if there is no room in the RTC to accept another posted write clearance to a field in the RTCSR0 MMR which is sourced in the 32 kHz domain.

Note that the WPEND status and WSYNC status for the RTCSR0 are encompassing values for the MMR as a whole. There is no distinction made in these status values as to how many interrupt source fields are being cleared at the same time by a single posted write to RTCSR0.

The RTCSR1 status register, in comparison, is a read only status register. RTCSR1 has no WSYNC or WPEND status values, as there are never any posted writes to the MMR.

Table 23. Bit Descriptions for RTCSR1

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	This field is reserved for ADI factory use only and reads back as all zeros.	0x0	R
13	WPENDTRM	Pending status of posted writes to RTCTRM. WPENDTRM indicates if a register write to RTCTRM is currently pending and awaiting execution. 0: the RTC can accept a new posted write to the RTCTRM MMR. 1: a previously posted write to RTCTRM is still awaiting execution, so no new posting to this MMR can be accepted.	0x0	R
12	WPENDALM1	Pending status of posted writes to RTCALM1. WPENDALM1 indicates if a register write to RTCALM1 is currently pending and awaiting execution. 0: the RTC can accept a new posted write to the RTCALM1 MMR. 1: a previously posted write to RTCALM1 is still awaiting execution, so no new posting to this MMR can be accepted.	0x0	R
11	WPENDALM0	Pending status of posted writes to RTCALM0. WPENDALM0 indicates if a register write to RTCALM0 is currently pending and awaiting execution. 0: the RTC can accept a new posted write to the RTCALM0 MMR. 1: a previously posted write to RTCALM0 is still awaiting execution, so no new posting to this MMR can be accepted.	0x0	R
10	WPENDCNT1	Pending status of posted writes to RTCCNT1. WPENDCNT1 indicates if a register write to RTCCNT1 is currently pending and awaiting execution. 0: the RTC can accept a new posted write to the RTCCNT1 MMR. 1: a previously posted write to RTCCNT1 is still awaiting execution, so no new posting to this MMR can be accepted.	0x0	R
9	WPENDCNT0	Pending status of posted writes to RTCCNT0. WPENDCNT0 indicates if a register write to RTCCNT0 is currently pending and awaiting execution. 0: the RTC can accept a new posted write to the RTCCNT0 MMR. 1: a previously posted write to RTCCNT0 is still awaiting execution, so no new posting to this MMR can be accepted.	0x0	R
8	WPENDSR0	Pending status of posted clearances of interrupt sources in RTCSR0. WPENDSR0 indicates if a register write to RTCSR0 is currently pending and awaiting execution. 0: the RTC can accept new posted clearances of interrupt sources located in the 32 kHz domain. No clearances are currently pending. 1: a previously posted clearance to interrupt sources in RTCSR0 maintained in the 32 kHz domain is still awaiting execution. Nevertheless, further posted clearances can still be accepted and accumulated along with the currently pending ones. This is because clearances posted at different times can cumulatively pend as one transaction, because the clearances are nondestructive to one another.	0x0	R

Bits	Bit Name	Description	Reset	Access
7	WPENDCR	Pending status of posted writes to RTCCR. WPENDCR indicates if a register write to RTCCR is currently pending and awaiting execution. 0: the RTC can accept a new posted write to RTCCR. 1: a previously posted write to RTCCR is still awaiting execution, so no new posting to this MMR can be accepted.	0x0	R
[6:3]	WERRCODE	Identifier for the source of a write synchronization error. WERRCODE is a sticky code that identifies the cause of a write synchronization error interrupt to the processor. WERRCODE sticks at the same time as WSYNCERRINT. When the CPU writes one to clear WSYNCERRINT, WERRCODE is also reset to its inert (no error) value. 0000: error due to a pending RTCCR write. 0001: error due to a pending RTCSR0 write. 0011: error due to a pending RTCCNT0 write. 0100: error due to a pending RTCCNT1 write. 0101: error due to a pending RTCALM0 write. 0110: error due to a pending RTCALM1 write. 0111: error due to a pending RTCTRM write. 1111: no error detected since WSYNCERRINT was last cleared.	0xF	R
[2:0]	RESERVED	This field is reserved for ADI factory use only and reads back as all zeros.	0x0	R

RTC Count 0 Register

Address: 0x4000260C, Reset: 0x0000, Name: RTCCNT0

RTCCNT0 contains the lower 16 bits of the RTC counter which maintains a real-time count in elapsed seconds.

The instantaneous value of RTCCNT0 can be read back by the CPU. The CPU can also redefine the value in this register. In this case, the RTC continues its real-time count from the redefined value.

Any write to RTCCNT0 remains pending until a corresponding write to RTCCNT1 is carried out by the CPU, so that the combined 32-bit count redefinition can be executed as a single transaction.

RTCCNT0 and RTCCNT1 can be written in either order, but paired, twin writes must be carried out by the CPU to have any effect on the RTC count.

A paired write to RTCCNT0 and RTCCNT1 (in whichever order) zeros the prescaler in the RTC and thus causes a redefinition of elapsed time by the CPU to align exactly with newly created 1 sec and 1 minute boundaries.

Such a redefinition also causes the RTC to create a trim boundary and initiate a new trim interval. When the RTC count is redefined by the CPU, no coincident trim adjustment of the count is carried out.

The RTC supports on the fly redefinition of RTCCNT0 and RTCCNT1 while CNTEN is active. Alternatively, the CPU can disable the RTC (by first making CNTEN inactive) while redefining these registers.

Table 24. Bit Descriptions for RTCCNT0

Bits	Bit Name	Description	Reset	Access
[15:0]	RTCCNT0	Lower 16 bits of the RTC real-time count.	0x0	RW

RTC Count 1 Register

Address: 0x40002610, Reset: 0x0000, Name: RTCCNT1

RTCCNT1 contains the upper 16 bits of the RTC counter which maintains a real-time count in elapsed seconds.

The instantaneous value of RTCCNT1 can be read back by the CPU. The CPU can also redefine the value in this register. In this case, the RTC continues its real-time count from the redefined value.

Any write to RTCCNT1 remains pending until a corresponding write to RTCCNT0 is carried out by the CPU, so that the combined 32-bit count redefinition can be executed as a single transaction. RTCCNT1 and RTCCNT0 can be written in either order, but paired, twin writes must be carried out by the CPU to have any effect on the RTC count.

A paired write to RTCCNT1 and RTCCNT0 (in whichever order) zeros the prescaler in the RTC and thus causes a redefinition of elapsed time by the CPU to align exactly with newly created 1 sec and 1 minute boundaries. Such a redefinition also causes the RTC to create a trim boundary and initiate a new trim interval. When the RTC count is redefined by the CPU, no coincident trim adjustment of the count is carried out.

The RTC supports on the fly redefinition of RTCCNT1 and RTCCNT0 while CNTEN is active. Alternatively, the CPU can disable the RTC (by first making CNTEN inactive) while redefining these registers.

Table 25. Bit Descriptions for RTCCNT1

Bits	Bit Name	Description	Reset	Access
[15:0]	RTCCNT1	Upper 16 bits of the RTC real-time count.	0x0	RW

RTC Alarm 0 Register

Address: 0x40002614, Reset: 0xFFFF, Name: RTCALM0

RTCALM0 contains the lower 16 bits of the RTC alarm value.

Any write to RTCALM0 remains pending until a corresponding write to RTCALM1 is carried out by the CPU, so that the combined 32-bit alarm redefinition can be executed as a single transaction. RTCALM0 and RTCALM1 can be written in either order, but paired, twin writes must be carried out by the CPU to have any effect on the RTC alarm.

Note that RTCALM0 can be written to regardless of whether ALMEN or CNTEN is active in the RTCCR register.

Table 26. Bit Descriptions for RTCALM0

Bits	Bit Name	Description	Reset	Access
[15:0]	RTCALM0	Lower 16 bits of the RTC alarm register. Note that the alarm register has a different reset value to the RTC count to avoid spurious alarms.	0xFFFF	RW

RTC Alarm 1 Register

Address: 0x40002618, Reset: 0xFFFF, Name: RTCALM1

RTCALM1 contains the upper 16 bits of the RTC alarm value.

Any write to RTCALM1 remains pending until a corresponding write to RTCALM0 is carried out by the CPU, so that the combined 32-bit alarm redefinition can be executed as a single transaction. RTCALM1 and RTCALM0 can be written in either order, but paired, twin writes must be carried out by the CPU to have any effect on the RTC alarm.

Note that RTCALM1 can be written to regardless of whether ALMEN or CNTEN is active in the RTCCR register.

Table 27. Bit Descriptions for RTCALM1

Bits	Bit Name	Description	Reset	Access
[15:0]	RTCALM1	Upper 16 bits of the RTC alarm register. Note that the alarm register has a different reset value to the RTC count to avoid spurious alarms.	0xFFFF	RW

RTC Trim Register

Address: 0x4000261C, Reset: 0x0018, Name: RTCTRM

RTCTRM contains the trim value and interval for a periodic adjustment of the RTC count value to track time with the required accuracy. Trimming can be enabled and disabled via the TRMEN bit in the RTCCR control register. For trimming to occur, the global enable for the RTC, CNTEN in the RTCCR register, must also be active.

Table 28. Bit Descriptions for RTCTRM

Bits	Bit Name	Description	Reset	Access
[5:4]	TRMIVL	Trim interval in units of seconds. TRMIVL specifies the interval at the end of which a periodic adjustment of TRMVAL seconds is made to the RTC count. Note that a trim boundary is created and a new trim interval is instigated if any of the following occurs: the RTC is reenabled from a disabled state via CNTEN in RTCCR; the trimming is reenabled from a disabled state via TRMEN in RTCCR; the RTCCNT0 and RTCCNT1 MMRs are redefined via paired writes, thus creating new 1 sec and 1 minute boundaries; or, the RTCTRM MMR is redefined. When a new trim interval is created under any of the above changes, the progression to date through any existing trim interval and any coincident trimming (if enabled) of the RTC count are immediately discarded. This is because all of the above events imply that the count starts/carries on from such a changing event with an alignment and value which are correct. Thus, a new trim interval is initiated in such circumstances, and only after the count in RTCCNT0 and RTCCNT1 has advanced by an increment equal to TRMIVL seconds after such an event is trimming actually carried out. Note that the period of trimming (that is, TRMIVL) is aligned to the most recent trim occurrence (that is, trim boundary for continuous trimming) or trim instigation (one of the events). There is no alignment to an absolute value or the toggling of a specific bit position within the RTCCNT0 or RTCCNT1 MMRs. 00: trim interval is 2^{14} sec. 01: trim interval is 2^{15} sec. 10: trim interval is 2^{16} sec. 11: trim interval is 2^{17} sec.	0x1	RW
3	TRMADD	Trim polarity. TRMADD specifies whether TRMVAL is a positive (additive) or negative (subtractive) adjustment of the RTC count. Note that a negative trim is implemented as a stall of the RTC count, whereas a positive trim results in an increment that is greater than the normal value of 1. The RTC count is never decremented as a result of trimming, because to do so means going back in time. See the description of the TRMVAL field of the RTCTRM MMR for more details. 1: add TRMVAL seconds to the RTC count at a period defined by TRMIVL. 0: subtract (by means of a stall) TRMVAL seconds from the RTC count at a period defined by TRMIVL.	0x1	RW
[2:0]	TRMVAL	Trim value in whole seconds to be added or subtracted from the RTC count at the end of a periodic interval. A trim adjustment of up to ±7 sec in the RTC count (at a period defined by TRMIVL) can be specified using TRMVAL. Note that positive (additive) trim adjustments result in an increment of more than one (time being skipped) to the RTC count. Negative (subtractive) adjustments result in a stall of the count, where the length of the stall equals the trim value to be subtracted from the count. The count is never decremented due to trimming, because such a course of action means going back in time, which is undesirable. Note that if an RTC interrupt event is configured to occur at a time that is coincidentally trimmed, the interrupt behavior is as follows. If the interrupt time is trimmed out (skipped) because of a positive trim, the interrupt issues at the end of the trim. For negative trims that coincidentally stall the RTC count at the target time for the interrupt event, the interrupt issues at the first occurrence of the target time. 000: make no adjustment to the RTC count at the end of a trim interval. 001: add or subtract one second (depending on TRMADD) to the RTC count at the end of a trim interval. 010: add or subtract two seconds (depending on TRMADD) to the RTC count at the end of a trim interval. 011: add or subtract three seconds (depending on TRMADD) to the RTC count at the end of a trim interval. 100: add or subtract four seconds (depending on TRMADD) to the RTC count at the end of a trim interval. 101: add or subtract five seconds (depending on TRMADD) to the RTC count at the end of a trim interval. 110: add or subtract six seconds (depending on TRMADD) to the RTC count at the end of a trim interval. 111: add or subtract seven seconds (depending on TRMADD) to the RTC count at the end of a trim interval.	0x0	RW

RTC Gateway Register

Address: 0x40002620, Reset: 0x0000, Name: RTCGWY

RTCGWY is a gateway MMR address through which the CPU can order exceptional action to be taken within the RTC by writing specific keys to RTCGWY. Currently, only one such key is supported.

Table 29. Bit Descriptions for RTCGWY

Bits	Bit Name	Description	Reset	Access
[15:0]	FLUSH_RTC	Software keyed cancellation of all RTC transactions, both pending and executing. The FLUSH_RTC key has the value 0xA2C5 and causes the RTC to flush (discard) all posted write transactions and to immediately stop any transaction that is currently executing. It is envisaged that this key is only used by the CPU when power loss to the device is imminent and the CPU wants to cleanly terminate activity across the power domain boundary of the RTC before the RTC activates its isolation barrier.	0x0	RW

Reserved—Analog Devices Factory Use Only Register

Address: 0x40002624, Reset: 0x0000, Name: RESERVED

This register is reserved for Analog Devices factory use only and reads back as all zeros.

Do not write to this register.

Table 30. Bit Descriptions for RESERVED

Bits	Bit Name	Description	Reset	Access
[15:0]	RESERVED	Analog Devices factory use only.	0x0000	RW

GENERAL-PURPOSE TIMERS

FEATURES

The ADuCM 350 has three identical general-purpose timers, each with a 16-bit count up/count down counter. The count up/count down counter can be clocked from one of four user-selectable clock sources. Any selected clock source can be scaled down using a prescaler of 16, 256, or 32,768. Additionally, two of the clocks can be scaled down using a prescaler of 4, while the other two clock sources can be used directly (prescaler of 1).

The timers have two modes of operation: free running and periodic. In free running mode, the counter decrements/increments from the maximum/minimum value until zero/full scale and starts again at the maximum/minimum value. In periodic mode, the counter decrements/increments from the value in the load register (GPTLD MMR) until zero/full scale and starts again at the value stored in the load register.

The value of a counter can be read at any time by accessing its value register (GPTVAL). This register is synchronized to the core clock; therefore, GPTVAL returns a slightly delayed timer value. Delay from synchronization logic can be avoided only if the timer is clocked from the core clock. In this case, a read from GPTAVAL returns the asynchronous (not synchronized) timer value directly. Reading GPTAVAL when the timer and core operate on different clocks is not supported, and the return value is undefined in this case.

Writing the timer control register (GPTCON) initiates the up/down counter. An interrupt is generated each time the value of the counter reaches 0 when counting down, or each time the counter value reaches full scale when counting up. An interrupt can be cleared by writing a 1 to the corresponding bit in the clear register (GPTCLRI). An interrupt is also set when a selected event occurs on the event bus.

In addition, there is a capture feature that is triggered by a selected interrupt source assertion. When triggered, the current timer value is copied to GPTCAP, and the timer continues to run. This feature can be used to determine the assertion of an event with increased accuracy. It can select one from the 16 different events as the trigger source for input to the block.

Each timer also provides a pulse-width modulator (PWM) function. This PWM function can be configured to idle in either Logic 0 or Logic 1 by writing the corresponding bit in GPTPCON. An optional match value can be written to GPTPMAT and enabled by writing another bit in GPTPCON. The PWM output is generated in one of two modes: toggle or match. In toggle mode, the PWM output toggles on timer zero/full scale, which provides a 50% duty cycle with a configurable period. Match mode provides a configurable duty cycle and a configurable period PWM output. In match mode, the PWM output starts in the idle state as configured in the GPTPCON register, is asserted when the timer and match values are equal, and is deasserted to the idle state again when the timer reaches zero/full scale.

BLOCK DIAGRAM

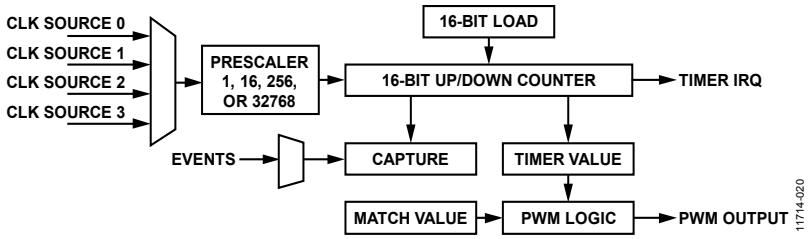


Figure 21. General-Purpose Timers Block Diagram

OPERATION

The general-purpose timer has two modes of operation: free running and periodic. In either timer mode, the PWM output can be used in toggle mode or match mode.

In free running mode, the timer is started by writing to the ENABLE bit of the GPTCON register. The timer increments from zero/full scale to full scale/zero if counting up/down. Full scale is 216 − 1, or 0xFFFF in hexadecimal format. Upon reaching full scale (or zero), a timeout interrupt occurs, and GPTSTA[0] is set. To clear GPTSTA[0], user code must write GPTCLRI[0]. The timer is reloaded with the maximum/minimum value when the timeout interrupt occurs. If GPTCON[7] is set, the timer is also reloaded when the GPTCLRI[0] register is written 1.

In periodic mode, load the initial GPTLD value before enabling the timer. The timer is started by writing the GPTCON enable bit. The counter value increments from the value stored in the GPTLD register to full scale or decrements from the value stored in the GPTLD register to 0, depending on the GPTCON[2] settings (count up/down). When the counter reaches full scale or zero, the timer generates an interrupt. The GPTLD is reloaded into the counter, and it continues counting up/down. Disable the timer prior to changing the GPTCON or GPTLD register. By default, the counter is reloaded automatically when generating the interrupt. If GPTCON[7] is set to 1, the counter is also reloaded when user code writes GPTCLRI.

To enable timing critical modifications to the load value, a nonsynchronized write address is provided at GPTALD. Writing GPTALD bypasses synchronization logic, providing finer grained control over the load value. If the GPTALD register is changed while the timer is enabled and running on a different clock than the core, undefined results may occur.

GPTSTA must be read prior to writing to any timer registers following the set or clear of enable. After GPTSTA[7] returns to 0, registers can be modified, which ensures that the timer finishes synchronizing timer control between the core and timer clock domains. The typical synchronization time is two timer clock periods. Any modification to Bit 2 or Bit 3 (UP and MOD) of the GPTCON register must be performed while the timer is disabled.

At any time, GPTVAL contains a valid value to be read, synchronized to the core clock. The GPTCON register enables the counter, selects the mode, selects the prescale value, and controls the event capture function.

The PWM can be configured to operate in either toggle mode or match mode. In either mode, using the PWM output requires selecting the appropriate mux settings in the GPIO control module.

In toggle mode, the PWM provides a 50% duty cycle output with a configurable period. The PWM output is inverted when a timeout interrupt is generated by the timer. The period is, therefore, defined by the selected clock and prescaler. If the timer is run in periodic mode (rather than free running), the PWM output also depends on the GPTLD value. Timeout events are evaluated at the end of a timer period, which is potentially much longer than a core clock period.

Match mode provides a configurable duty cycle and configurable period PWM output. Like toggle mode, the period is defined by the selected clock and the prescaler, as well as GPTLD (when the timer is run in periodic mode). The duty cycle is defined by the GPTPMAT value. When the counter value of the timer is equal to the value stored in GPTPMAT, the PWM output is asserted. It remains asserted until the timer reaches zero/full scale and is then deasserted. Match and timeout events are evaluated at the end of a timer period, which is potentially much longer than a core clock period. Match events take priority over timeout events when triggered on the same cycle; therefore, a 100% duty cycle PWM can be configured by setting GPTPMAT equal to GPTLD when running in periodic mode, or to zero/full scale when in free running mode. A 0% duty cycle is only possible when running in periodic mode and can be configured by setting GPTPMAT outside of the range of the counter of the timer (GPTLD + 1 when counting down; GPTLD − 1 when counting up).

For a given clock source, prescaler, and GPTLD value, the PWM period in toggle mode is twice that of the period in match mode, due to the fact that toggle mode inverts the PWM output just once each timer counter period; whereas match mode both asserts and deasserts the PWM output. Writes to the GPTPCON and GPTPMAT registers while the timer is running are supported. However, the PWM output does not reflect changes to GPTPCON until the next match or timeout event, which enables modifying the PWM behavior while maintaining a deterministic PWM duty cycle. It is possible to force the PWM output to match newly written settings by writing to GPTCLRI or by disabling/enabling the timer.

In the GPIOs section and the Pins section, the general-purpose timer outputs are referred to as TOUTA, TOUTB, and TOUTC.

Clock Select

There are four clocks available for each of the three timers. The selected clock is used, along with a prescaler, to control the frequency of the counter logic of the timer. The available clocks are selected using GPTCON[6:5]. For proper general-purpose timer operation, the frequency of PCLK must be equal to or greater than four times the postprescaler clock frequency. For example, with a PCLK frequency of 16 MHz, HFOSC (a 16 MHz internal oscillator) is the selected clock source, and the prescaler must be set to a minimum of 4.

For this reason, when selecting PCLK or HFOSC as the clock source, the minimal prescaler value (typically 1) is internally adjusted to 4. Selecting LFXTAL as the clock source may also require enabling it in the clock control module if it is not already enabled (LFXTALEN bit in OSCCTRL register). Clock configuration settings are part of the clock control module and are detailed in the System Clocks section.

The timer modules can get starved of clock if the clock source mux is set to a clock that is not (or no longer) present. In that state, the mux cannot be changed back (that is, the timer domain never synchronizes). To avoid this, always check clock status before setting timer mux to anything other than PCLK and then set mux back to PCLK before disabling the timer after its use. A clock must be verified to be present before switching the timer mux to that clock.

Table 31. Clock Select

Clock Select	GPT0/TOUTA Clock Source	GPT1/TOUTB Clock Source	GPT2/TOUTC Clock Source
0	PCLK	PCLK	PCLK
1	HFOSC	HFOSC	HFOSC
2	LFOSC	LFOSC	LFOSC
3	LFXTAL	LFXTAL	LFXTAL

Capture Event Function

There are 16 interrupt events that can be captured by the general-purpose timers. Any one of the 16 events associated with a general-purpose timer can cause a capture of the 16-bit GPTVAL register into the 16-bit GPTCAP register. GPTCON has a 4-bit field selecting which of the 16 events to capture.

When the selected IRQ occurs, the GPTVAL register is copied into the GPTCAP register. Bit 1 of the GPTSTA register is set. The IRQ is cleared by writing 1 to Bit 1 of the GPTCLRI register. The GPTCAP register holds its value and cannot be overwritten until Bit 1 of the GPTCLRI register is written with a 1.

Table 32. Capture Event Function

Event Select Bits (Bits[11:8] of GPTxCON)	GPT0 Capture Source	GPT1 Capture Source	GPT2 Capture Source
0	Wake-up timer	UART	External Interrupt 0
1	External Interrupt 0	SPI0	External Interrupt 1
2	External Interrupt 1	SPI1	External Interrupt 2
3	External Interrupt 2	SPIH	External Interrupt 3
4	External Interrupt 3	I^2C slave	External Interrupt 4
5	External Interrupt 4	I^2C master	External Interrupt 5
6	External Interrupt 5	I^2S	External Interrupt 6
7	External Interrupt 6	USB	External Interrupt 7
8	External Interrupt 7	USB DMA	External Interrupt 8
9	External Interrupt 8	AFE—capture	AFE—generation
10	Reserved	Beeper	CapTouch
11	CapTouch	DMA error	RTC
12	WDT	DMA done (any channel)	AFE—command FIFO
13	Flash controller	General-purpose flash controller	AFE—data FIFO
14	Timer 1	Timer 0	Timer 0
15	Timer 2	Timer 2	Timer 1

GENERAL-PURPOSE TIMERS MEMORY MAPPED REGISTERS

Each general-purpose timer consists of 10 MMRs, GPT0 MMRs begin at Address 0x40000000, GPT1 MMRs begin at Address 0x40000400, and GPT2 MMRs begin at Address 0x40000800.

General-Purpose Timers Register Map

Table 33. General-Purpose Timers Register Summary

Address	Name	Description	Reset	RW
0x40000000	GPT0LD	16-bit load value	0x0000	RW
0x40000004	GPT0VAL	16-bit timer value	0x0000	R
0x40000008	GPT0CON	Control	0x000A	RW
0x4000000C	GPT0CLRI	Clear interrupt	0x0000	W
0x40000010	GPT0CAP	Capture	0x0000	R
0x40000014	GPT0ALD	16-bit load value, asynchronous	0x0000	RW
0x40000018	GPT0AVAL	16-bit timer value, asynchronous	0x0000	R
0x4000001C	GPT0STA	Status	0x0000	R
0x40000020	GPT0PCON	PWM control register	0x0000	RW
0x40000024	GPT0PMAT	PWM match value	0x0000	RW
0x40000400	GPT1LD	16-bit load value	0x0000	RW
0x40000404	GPT1VAL	16-bit timer value	0x0000	R
0x40000408	GPT1CON	Control	0x000A	RW
0x4000040C	GPT1CLRI	Clear interrupt	0x0000	W
0x40000410	GPT1CAP	Capture	0x0000	R
0x40000414	GPT1ALD	16-bit load value, asynchronous	0x0000	RW
0x40000418	GPT1AVAL	16-bit timer value, asynchronous	0x0000	R
0x4000041C	GPT1STA	Status	0x0000	R
0x40000420	GPT1PCON	PWM control register	0x0000	RW
0x40000424	GPT1PMAT	PWM match value	0x0000	RW
0x40000800	GPT2LD	16-bit load value	0x0000	RW
0x40000804	GPT2VAL	16-bit timer value	0x0000	R
0x40000808	GPT2CON	Control	0x000A	RW
0x4000080C	GPT2CLRI	Clear interrupt	0x0000	W
0x40000810	GPT2CAP	Capture	0x0000	R
0x40000814	GPT2ALD	16-bit load value, asynchronous	0x0000	RW
0x40000818	GPT2AVAL	16-bit timer value, asynchronous	0x0000	R
0x4000081C	GPT2STA	Status	0x0000	R
0x40000820	GPT2PCON	PWM control register	0x0000	RW
0x40000824	GPT2PMAT	PWM match value	0x0000	RW

REGISTER SUMMARY: GPT0

Table 34. General-Purpose Timer 0 Register Summary

Address	Name	Description	Reset	RW
0x40000000	GPT0LD	16-bit load value	0x0000	RW
0x40000004	GPT0VAL	16-bit timer value	0x0000	R
0x40000008	GPT0CON	Control	0x000A	RW
0x4000000C	GPT0CLRI	Clear interrupt	0x0000	W
0x40000010	GPT0CAP	Capture	0x0000	R
0x40000014	GPT0ALD	16-bit load value, asynchronous	0x0000	RW
0x40000018	GPT0AVAL	16-bit timer value, asynchronous	0x0000	R
0x4000001C	GPT0STA	Status	0x0000	R
0x40000020	GPT0PCON	PWM control register	0x0000	RW
0x40000024	GPT0PMAT	PWM match value	0x0000	RW

16-Bit Load Value Register

Address: 0x40000000, Reset: 0x0000, Name: GPT0LD

Table 35. Bit Descriptions for GPT0LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value. The up/down counter is periodically loaded with this value if periodic mode is selected (GPTCON[3] = 1). LOAD writes during up/down counter timeout events are delayed until the event has passed.	0x0	RW

16-Bit Timer Value Register

Address: 0x40000004, Reset: 0x0000, Name: GPT0VAL

Table 36. Bit Descriptions for GPT0VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VAL	Current count. Reflects the current up/down counter value. Value delayed two PCLK cycles due to clock synchronizers.	0x0	R

Control Register

Address: 0x40000008, Reset: 0x000A, Name: GPT0CON

Table 37. Bit Descriptions for GPT0CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED		0x0	R
12	EVENTEN	Event select. Used to enable and disable the capture of events. Used in conjunction with the EVENT select range: when a selected event occurs, the current value of the up/down counter is captured in GPTCAP. 0: events are not captured. 1: events are captured.	0x0	RW
[11:8]	EVENT	Event select range. Timer event select range (0 to 15).	0x0	RW
7	RLD	Reload control. This bit allows the user to select whether the up/down counter is reset only on a timeout event or also when GPTCLRI[0] is set. 1: resets the up/down counter when GPTCLRI[0] is set. 0: up/down counter is only reset on a timeout event.	0x0	RW
[6:5]	CLK	Clock select. Used to select a timer clock from the four available clock sources. 00: select CLK Source 0(default). 01: select CLK Source 1. 10: select CLK Source 2. 11: select CLK Source 3.	0x0	RW
4	ENABLE	Timer enable. Used to enable and disable the timer. Clearing this bit resets the timer, including the GPTVAL register. 0: timer is disabled (default). 1: timer is enabled.	0x0	RW
3	MOD	Timer mode. This bit is used to control whether the timer runs in periodic or free running mode. In periodic mode the up/down counter starts at the defined LOAD value; in free running mode the up/down counter starts at 0x0000 or 0xFFFF depending on whether the timer is counting up or down. 1: timer runs in periodic mode (default). 0: timer runs in free running mode.	0x1	RW
2	UP	Count up. Used to control whether the timer increments (counts up) or decrements (counts down) the up/down counter. 1: timer is set to count up. 0: timer is set to count down (default).	0x0	RW

Bits	Bit Name	Description	Reset	Access
[1:0]	PRE	Prescaler. Controls the prescaler division factor applied to the selected clock of the timer. If CLK Source 0 or CLK Source 1 are selected, Prescaler Value 0 means divide by 4; otherwise, it means divide by 1. 00: source clock/(1 or 4). 01: source clock/16. 10: source clock/256. 11: source clock/32,768.	0x2	RW

Clear Interrupt Register

Address: 0x4000000C, Reset: 0x0000, Name: GPT0CLRI

Table 38. Bit Descriptions for GPT0CLRI

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED		0x0	R
1	CAP	Clear captured event interrupt. This bit is used to clear a capture event interrupt. 1: clear the capture event interrupt. 0: no effect.	0x0	W1C
0	TMOUT	Clear timeout interrupt. This bit is used to clear a timeout interrupt. 1: clears the timeout interrupt. 0: no effect.	0x0	W1C

Capture Register

Address: 0x40000010, Reset: 0x0000, Name: GPT0CAP

Table 39. Bit Descriptions for GPT0CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	CAP	16-bit captured value. GPTCAP holds its value until GPTCLRI[1] is set by user code. GPTCAP is not overwritten even if another event occurs without writing to GPTCLRI[1].	0x0	R

16-Bit Load Value, Asynchronous Register

Address: 0x40000014, Reset: 0x0000, Name: GPT0ALD

Only use when a synchronous clock source is selected (GPTCON[6:5] = 00)

Table 40. Bit Descriptions for GPT0ALD

Bits	Bit Name	Description	Reset	Access
[15:0]	ALOAD	Load value, asynchronous. The up/down counter is periodically loaded with this value if periodic mode is selected (GPTCON[3] = 1). Writing ALOAD takes advantage of having the timer run on PCLK by bypassing clock synchronization logic otherwise required.	0x0	RW

16-Bit Timer Value, Asynchronous Register

Address: 0x40000018, Reset: 0x0000, Name: GPT0AVAL

Only use when a synchronous clock source is selected (GPTCON[6:5] = 00)

Table 41. Bit Descriptions for GPT0AVAL

Bits	Bit Name	Description	Reset	Access
[15:0]	AVAL	Counter value. Reflects the current up/down counter value. Reading AVAL takes advantage of having the timer run on PCLK by bypassing clock synchronization logic otherwise required.	0x0	R

Status Register

Address: 0x4000001C, Reset: 0x0000, Name: GPT0STA

Table 42. Bit Descriptions for GPT0STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	PDOK	GPTCLRI synchronization. This bit is set automatically when the user sets GPTCLRI[0] = 1. It is cleared automatically when the clear interrupt request has crossed clock domains and taken effect in the timer clock domain. 1: GPTCLRI[0] is being updated in the timer clock domain. 0: the interrupt is cleared in the timer clock domain.	0x0	R
6	BUSY	Timer busy. This bit informs the user that a write to GPTCON is still crossing into the timer clock domain. Check this bit after writing GPTCON, and suppress further writes until this bit is cleared. 0: timer ready to receive commands to GPTCON. 1: timer not ready to receive commands to GPTCON.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: no capture event is pending. 1: a capture event is pending.	0x0	R
0	TMOUT	Timeout event occurred. This bit set automatically when the value of the counter reaches zero while counting down or reaches full scale when counting up. This bit is cleared when GPTCLRI[0] is set by the user. 0: no timeout event has occurred. 1: a timeout event has occurred.	0x0	R

PWM Control Register

Address: 0x40000020, Reset: 0x0000, Name: GPT0PCON

Table 43. Bit Descriptions for GPT0PCON

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	R
1	IDLE_STATE	PWM idle state. This bit is used to set the PWM idle state. 0: PWM idles low. 1: PWM idles high.	0x0	RW
0	MATCH_EN	PWM match enabled. This bit is used to control PWM operational mode. 0: PWM in toggle mode. 1: PWM in match mode.	0x0	RW

PWM Match Value Register

Address: 0x40000024, Reset: 0x0000, Name: GPT0PMAT

Table 44. Bit Descriptions for GPT0PMAT

Bits	Bit Name	Description	Reset	Access
[15:0]	MATCH_VAL	PWM match value. The value is used when the PWM is operating in match mode. The PWM output is asserted when the up/down counter is equal to this match value. PWM output is deasserted again when a timeout event occurs. If the match value is never reached, or occurs simultaneous to a timeout event, the PWM output remains idle.	0x0	RW

REGISTER SUMMARY: GPT1

Table 45. General-Purpose Timer1 Register Summary

Address	Name	Description	Reset	RW
0x40000400	GPT1LD	16-bit load value	0x0000	RW
0x40000404	GPT1VAL	16-bit timer value	0x0000	R
0x40000408	GPT1CON	Control	0x000A	RW
0x4000040C	GPT1CLRI	Clear interrupt	0x0000	W
0x40000410	GPT1CAP	Capture	0x0000	R
0x40000414	GPT1ALD	16-bit load value, asynchronous	0x0000	RW
0x40000418	GPT1AVAL	16-bit timer value, asynchronous	0x0000	R
0x4000041C	GPT1STA	Status	0x0000	R
0x40000420	GPT1PCON	PWM control register	0x0000	RW
0x40000424	GPT1PMAT	PWM match value	0x0000	RW

16-Bit Load Value Register

Address: 0x40000400, Reset: 0x0000, Name: GPT1LD

Table 46. Bit Descriptions for GPT1LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value. The up/down counter is periodically loaded with this value if periodic mode is selected (GPTCON[3] = 1). LOAD writes during up/down counter timeout events are delayed until the event has passed.	0x0	RW

16-Bit Timer Value Register

Address: 0x40000404, Reset: 0x0000, Name: GPT1VAL

Table 47. Bit Descriptions for GPT1VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VAL	Current count. Reflects the current up/down counter value. Value delayed two PCLK cycles due to clock synchronizers.	0x0	R

Control Register

Address: 0x40000408, Reset: 0x000A, Name: GPT1CON

Table 48. Bit Descriptions for GPT1CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED		0x0	R
12	EVENTEN	Event select. Used to enable and disabling the capture of events. Used in conjunction with the EVENT select range: when a selected event occurs the current value of the up/down counter is captured in GPTCAP. 0: events are not captured. 1: events are captured.	0x0	RW
[11:8]	EVENT	Event select range. Timer event select range (0 to 15).	0x0	RW
7	RLD	Reload control. This bit allows the user to select whether the up/down counter is reset only on a timeout event or also when Bit 0 of GPTCLRI is set. 1: Resets the up/down counter when GPTCLRI[0] is set. 0: Up/Down counter is only reset on a timeout event.	0x0	RW
[6:5]	CLK	Clock select. Used to select a timer clock from the four available clock sources. 00: select CLK Source 0 (default). 01: select CLK Source 1. 10: select CLK Source 2. 11: select CLK Source 3.	0x0	RW

Bits	Bit Name	Description	Reset	Access
4	ENABLE	Timer enable. Used to enable and disable the timer. Clearing this bit resets the timer, including the GPTVAL register. 0: timer is disabled (default). 1: timer is enabled.	0x0	RW
3	MOD	Timer mode. This bit is used to control whether the timer runs in periodic or free running mode. In periodic mode the up/down counter starts at the defined LOAD value; in free running mode the up/down counter starts at 0x0000 or 0xFFFF depending on whether the timer is counting up or down. 1: timer runs in periodic mode (default). 0: timer runs in free running mode.	0x1	RW
2	UP	Count up. Used to control whether the timer increments (counts up) or decrements (counts down) the up/down counter. 1: timer is set to count up. 0: timer is set to count down (default).	0x0	RW
[1:0]	PRE	Prescaler. Controls the prescaler division factor applied to the selected clock of the timer. If CLK Source 0 or CLK Source 1 are selected, Prescaler Value 0 means divide by 4; otherwise, it means divide by 1. 00: source clock/(1 or 4). 01: source clock/16. 10: source clock/256. 11: source clock/32,768.	0x2	RW

Clear Interrupt Register

Address: 0x4000040C, Reset: 0x0000, Name: GPT1CLRI

Table 49. Bit Descriptions for GPT1CLRI

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	R
1	CAP	Clear captured event interrupt. This bit is used to clear a capture event interrupt. 1: clear the capture event interrupt. 0: no effect.	0x0	W1C
0	TMOUT	Clear timeout interrupt. This bit is used to clear a timeout interrupt. 1: clears the timeout interrupt. 0: no effect.	0x0	W1C

Capture Register

Address: 0x40000410, Reset: 0x0000, Name: GPT1CAP

Table 50. Bit Descriptions for GPT1CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	CAP	16-bit captured value. GPTCAP holds its value until GPTCLRI[1] is set by user code. GPTCAP is not overwritten even if another event occurs without writing to GPTCLRI[1].	0x0	R

16-Bit Load Value, Asynchronous Register

Address: 0x40000414, Reset: 0x0000, Name: GPT1ALD

Only use when a synchronous clock source is selected (GPTCON[6:5] = 00)

Table 51. Bit Descriptions for GPT1ALD

Bits	Bit Name	Description	Reset	Access
[15:0]	ALOAD	Load value, asynchronous. The up/down counter is periodically loaded with this value if periodic mode is selected (GPTCON[3] = 1). Writing ALOAD takes advantage of having the timer run on PCLK by bypassing clock synchronization logic otherwise required.	0x0	RW

16-Bit Timer Value, Asynchronous Register

Address: 0x40000418, Reset: 0x0000, Name: GPT1AVAL

Only use when a synchronous clock source is selected (GPTCON[6:5] = 00)

Table 52. Bit Descriptions for GPT1AVAL

Bits	Bit Name	Description	Reset	Access
[15:0]	AVAL	Counter value. Reflects the current up/down counter value. Reading AVAL takes advantage of having the timer run on PCLK by bypassing clock synchronization logic otherwise required.	0x0	R

Status Register

Address: 0x4000041C, Reset: 0x0000, Name: GPT1STA

Table 53. Bit Descriptions for GPT1STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED		0x0	R
7	PDOK	GPTCLRI synchronization. This bit is set automatically when the user sets GPTCLRI[0] = 1. It is cleared automatically when the clear interrupt request has crossed clock domains and taken effect in the timer clock domain. 1: GPTCLRI[0] is being updated in the timer clock domain. 0: the interrupt is cleared in the timer clock domain.	0x0	R
6	BUSY	Timer busy. This bit informs the user that a write to GPTCON is still crossing into the timer clock domain. This bit is checked after writing GPTCON and further writes are suppressed until this bit is cleared. 0: timer ready to receive commands to GPTCON. 1: timer not ready to receive commands to GPTCON.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: no capture event is pending. 1: a capture event is pending.	0x0	R
0	TMOUT	Timeout event occurred. This bit is set automatically when the value of the counter reaches zero while counting down or reaches full scale when counting up. This bit is cleared when GPTCLRI[0] is set by the user. 0: no timeout event has occurred. 1: a timeout event has occurred.	0x0	R

PWM Control Register

Address: 0x40000420, Reset: 0x0000, Name: GPT1PCON

Table 54. Bit Descriptions for GPT1PCON

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	R
1	IDLE_STATE	PWM idle state. This bit is used to set the PWM idle state. 0: PWM idles low. 1: PWM idles high.	0x0	RW
0	MATCH_EN	PWM match enabled. This bit is used to control PWM operational mode. 0: PWM in toggle mode. 1: PWM in match mode.	0x0	RW

PWM Match Value Register

Address: 0x40000424, Reset: 0x0000, Name: GPT1PMAT

Table 55. Bit Descriptions for GPT1PMAT

Bits	Bit Name	Description	Reset	Access
[15:0]	MATCH_VAL	PWM match value. The value is used when the PWM is operating in match mode. The PWM output is asserted when the up/down counter is equal to this match value. PWM output is deasserted again when a timeout event occurs. If the match value is never reached, or occurs simultaneous to a timeout event, the PWM output remains idle.	0x0	RW

REGISTER SUMMARY: GPT2

Table 56. General-Purpose Timer2 Register Summary

Address	Name	Description	Reset	RW
0x40000800	GPT2LD	16-bit load value	0x0000	RW
0x40000804	GPT2VAL	16-bit timer value	0x0000	R
0x40000808	GPT2CON	Control	0x000A	RW
0x4000080C	GPT2CLRI	Clear interrupt	0x0000	W
0x40000810	GPT2CAP	Capture	0x0000	R
0x40000814	GPT2ALD	16-bit load value, asynchronous	0x0000	RW
0x40000818	GPT2AVAL	16-bit timer value, asynchronous	0x0000	R
0x4000081C	GPT2STA	Status	0x0000	R
0x40000820	GPT2PCON	PWM control register	0x0000	RW
0x40000824	GPT2PMAT	PWM match value	0x0000	RW

16-Bit Load Value Register

Address: 0x40000800, Reset: 0x0000, Name: GPT2LD

Table 57. Bit Descriptions for GPT2LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value. The up/down counter is periodically loaded with this value if periodic mode is selected (GPTCON[3] = 1). LOAD writes during up/down counter timeout events are delayed until the event has passed.	0x0	RW

16-Bit Timer Value Register

Address: 0x40000804, Reset: 0x0000, Name: GPT2VAL

Table 58. Bit Descriptions for GPT2VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VAL	Current count. Reflects the current up/down counter value. Value delayed two PCLK cycles due to clock synchronizers.	0x0	R

Control Register

Address: 0x40000808, Reset: 0x000A, Name: GPT2CON

Table 59. Bit Descriptions for GPT2CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED		0x0	R
12	EVENTEN	Event select. Used to enable and disabling the capture of events. Used in conjunction with the EVENT select range: when a selected event occurs the current value of the up/down counter is captured in GPTCAP. 0: events are not captured. 1: events are captured.	0x0	RW
[11:8]	EVENT	Event select range. Timer event select range (0 to 15).	0x0	RW
7	RLD	Reload control. This bit allows the user to select whether the up/down counter is reset only on a timeout event or also when Bit 0 of GPTCLRI is set. 1: resets the up/down counter when GPTCLRI[0] is set. 0: up/down counter is only reset on a timeout event.	0x0	RW
[6:5]	CLK	Clock select. Used to select a timer clock from the four available clock sources. 00: select CLK Source 0 (default). 01: select CLK Source 1. 10: select CLK Source 2. 11: select CLK Source 3.	0x0	RW
4	ENABLE	Timer enable. Used to enable and disable the timer. Clearing this bit resets the timer, including the GPTVAL register. 0: timer is disabled (default). 1: timer is enabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
3	MOD	Timer mode. This bit is used to control whether the timer runs in periodic or free running mode. In periodic mode the up/down counter starts at the defined LOAD value; in free running mode the up/down counter starts at 0x0000 or 0xFFFF depending on whether the timer is counting up or down. 1: timer runs in periodic mode (default). 0: timer runs in free running mode.	0x1	RW
2	UP	Count up. Used to control whether the timer increments (counts up) or decrements (counts down) the up/down counter. 1: timer is set to count up. 0: timer is set to count down (default).	0x0	RW
[1:0]	PRE	Prescaler. Controls the prescaler division factor applied to the selected clock of the timer. If CLK Source 0 or CLK Source 1 are selected, Prescaler Value 0 means divide by 4; otherwise, it means divide by 1. 00: source clock/(1 or 4). 01: source clock/16. 10: source clock/256. 11: source clock/32,768.	0x2	RW

Clear Interrupt Register

Address: 0x4000080C, Reset: 0x0000, Name: GPT2CLRI

Table 60. Bit Descriptions for GPT2CLRI

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	R
1	CAP	Clear captured event interrupt. This bit is used to clear a capture event interrupt. 1: clear the capture event interrupt. 0: no effect.	0x0	W1C
0	TMOUT	Clear timeout interrupt. This bit is used to clear a timeout interrupt. 1: clears the timeout interrupt. 0: no effect.	0x0	W1C

Capture Register

Address: 0x40000810, Reset: 0x0000, Name: GPT2CAP

Table 61. Bit Descriptions for GPT2CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	CAP	16-bit captured value. GPTCAP holds its value until GPTCLRI[1] is set by user code. GPTCAP is not overwritten even if another event occurs without writing to GPTCLRI[1].	0x0	R

16-Bit Load Value, Asynchronous Register

Address: 0x40000814, Reset: 0x0000, Name: GPT2ALD

Only use when a synchronous clock source is selected (GPTCON[6:5] = 00).

Table 62. Bit Descriptions for GPT2ALD

Bits	Bit Name	Description	Reset	Access
[15:0]	ALOAD	Load value, asynchronous. The up/down counter is periodically loaded with this value if periodic mode is selected (GPTCOM[3] = 1). Writing ALOAD takes advantage of having the timer run on PCLK by bypassing clock synchronization logic otherwise required.	0x0	RW

16-Bit Timer Value, Asynchronous Register

Address: 0x40000818, Reset: 0x0000, Name: GPT2AVAL

Only use when a synchronous clock source is selected (GPTCON[6:5] = 00).

Table 63. Bit Descriptions for GPT2AVAL

Bits	Bit Name	Description	Reset	Access
[15:0]	AVAL	Counter value. Reflects the current up/down counter value. Reading AVAL takes advantage of having the timer run on PCLK by bypassing clock synchronization logic otherwise required.	0x0	R

Status Register

Address: 0x4000081C, Reset: 0x0000, Name: GPT2STA

Table 64. Bit Descriptions for GPT2STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED		0x0	R
7	PDOK	GPTCLRI synchronization. This bit is set automatically when the user sets GPTCLRI[0] = 1. It is cleared automatically when the clear interrupt request has crossed clock domains and taken effect in the timer clock domain. 1: GPTCLRI[0] is being updated in the timer clock domain. 0: the interrupt is cleared in the timer clock domain.	0x0	R
6	BUSY	Timer busy. This bit informs the user that a write to GPTCON is still crossing into the timer clock domain. This bit is checked after writing GPTCON and further writes are suppressed until this bit is cleared. 0: timer ready to receive commands to GPTCON. 1: timer not ready to receive commands to GPTCON.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: no capture event is pending. 1: a capture event is pending.	0x0	R
0	TMOUT	Timeout event occurred. This bit set automatically when the value of the counter reaches zero while counting down or reaches full scale when counting up. This bit is cleared when GPTCLRI[0] is set by the user. 0: no timeout event has occurred. 1: a timeout event has occurred.	0x0	R

PWM Control Register

Address: 0x40000820, Reset: 0x0000, Name: GPT2PCON

Table 65. Bit Descriptions for GPT2PCON

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	R
1	IDLE_STATE	PWM idle state. This bit is used to set the PWM idle state. 0: PWM idles low. 1: PWM idles high.	0x0	RW
0	MATCH_EN	PWM match enabled. This bit is used to control PWM operational mode. 0: PWM in toggle mode. 1: PWM in match mode.	0x0	RW

PWM Match Value Register

Address: 0x40000824, Reset: 0x0000, Name: GPT2PMAT

Table 66. Bit Descriptions for GPT2PMAT

Bits	Bit Name	Description	Reset	Access
[15:0]	MATCH_VAL	PWM match value. The value is used when the PWM is operating in match mode. The PWM output is asserted when the up/down counter is equal to this match value. PWM output is deasserted again when a timeout event occurs. If the match value is never reached, or occurs simultaneous to a timeout event, the PWM output remains idle.	0x0	RW

WAKE-UP TIMER (WUT)

FEATURES

The wake-up timer consists of a 32-bit counter clocked from the 32 kHz external crystal (LFXTAL), 32 kHz internal oscillator (LFOSC), or peripheral clock (PCLK). The selected clock source can be scaled down using a prescaler of 1, 16, 256, or 32,768.

The WUT continues to run when the core clock is disabled as long as it is not using PCLK as its source clock. It can be used in free running or periodic mode. In free running mode, the timer counts from 0x00000000 to 0xFFFFFFFF and starts again from 0x00000000. In periodic mode, the timer counts from 0x00000000 to T2WUFD.

In addition, the wake-up timer has four specific time fields to compare with the wake-up counter: T2WUFA, T2WUFB, T2WUFC, and T2WUFD. Any of these time fields can generate an interrupt or wake-up signal. When in free running mode, T2WUFB, T2WUFC, and T2WUFD need to be reconfigured in the software to generate a periodic interrupt. The fourth comparator, T2WUFA, is automatically incremented with the programmable 12-bit interval register.

BLOCK DIAGRAM

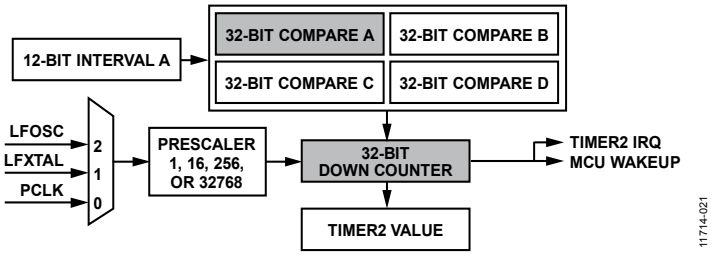


Figure 22. Wake-Up Timer Block Diagram

OPERATION

The wake-up timer comparator registers must be configured before starting the timer. The timer is started by writing to the control enable bit. The timer increments until the value reaches full scale in free running mode or when T2WUFD matches the wake-up value, T2VAL.

The T2VAL register can be read at any time, it reflects the current value of the counter. Because the wake-up timer is 32 bits but resides on a 16-bit bus, two bus reads are required to obtain the 32 bits. There are separate addresses for the upper (T2VAL1) and lower (T2VAL0) 16 bits of the wake-up timer. When the lower 16 bits are addressed and read, the upper 16 bits are latched and held in a separate register to be read later. The entire T2VAL register (upper and lower) remains frozen until the upper 16 bits are read. The control bit, freeze (Bit 3 of the T2CON register), must be set to freeze the T2VAL register between the lower and upper reads.

INTERVAL REGISTER

T2INC is a 12-bit interval register. When a new value is written in the T2INC register, Bits[16:5] of the internal 32-bit compare registers (T2WUFA0/T2WUFA1) are loaded with the new T2INC value. These 32-bit compare registers are automatically incremented with the contents of the T2INC register (shifted by 5) each time the wake-up counter reaches the value in the compare registers. This is provided that the new compare is less than the T2WUFD value if in periodic mode or 0xFFFFFFFF if in free running mode. If the new compare value is greater than these limits, it is recalculated as follows:

In free-running mode,

$$New\ T2WUFA = Old\ T2WUFA + T2INC - 0xFFFFFFFF$$

In periodic mode,

$$New\ T2WUFA = Old\ T2WUFA + T2INC - T2WUFD$$

The maximum programmable interval is 136 years.

To modify the interval T2INC value, the timer must be stopped so that the interval register can be loaded in the compare registers.

T2INC can also be modified while the timer is running, provided that the STOP_WUFA bit in the control register is set. The new T2INC value takes effect after the next wake-up interrupt. If the timer is running, update the T2INC sequence as follows:

1. User writes to STOP_WUFA bit.
2. User updates T2INC.
3. User clears STOP_WUFA bit.

The internal compare register, T2WUFA, can also be updated while the timer is running, provided that the STOP_WUFA bit in the control register is set. The compare register is updated four WUT_CLK cycles after T2WUFA1 is written to. This mechanism is used only when the timer is enabled. When the timer is disabled, the compare register is updated from the T2INC register.

Compare Fields Registers

T2WUFB, T2WUFC, and T2WUFD are 32-bit values programmed by the user in T2WUFx0 and T2WUFx1 registers (where x = B, C, or D). T2WUFD contains the load value when the wake-up timer is configured in periodic mode.

In free running mode, T2WUFB, T2WUFC, and T2WUFD can be written to at any time; however, the corresponding interrupt enable in the T2IEN register must be disabled. After the register is updated, the interrupt can be reenabled. In periodic mode, this is only applicable to T2WUFB and T2WUFC. The T2WUFD register can only be written to when the timer is disabled.

Interrupts/Wake-Up Signals

An IRQ is generated when the counter value corresponds to any of the compare points or to full scale in free running mode. The timer continues counting or is reset to 0.

The WUT generates five maskable interrupts that are enabled in the T2IEN register. An IRQ can be cleared by setting the corresponding bit in the T2CLRI register.

Note that it takes two 32 kHz clock cycles for the interrupt clear to take effect, enabling or disabling the timer. During that time, the system must not be placed into any of the power-down modes that stops PCLK. The IRQCRY and PDOK bits in T2STA indicate when the device can be placed in power-down mode. In addition, the system must not perform a second write to the T2CON within 32 kHz clock cycles of when it is synchronizing the first T2CON write between PCLK and 32 kHz clock.

WUT MEMORY MAPPED REGISTERS

WUT Register Summary

Table 67. WUT Register Summary

Address	Name	Description	Reset	RW
0x40002500.	T2VAL0	Current count value, LSB	R	0x0000
0x40002504	T2VAL1	Current count value, MSB	R	0x0000
0x40002508	T2CON	Control register	RW	0x0040
0x4000250C	T2INC	12-bit interval register for Wake-Up Field A	RW	0x0C8
0x40002510	T2WUFB0	Wake-Up Field B, LSB	RW	0x1FFF
0x40002514	T2WUFB1	Wake-Up Field B, MSB	RW	0x0000
0x40002518	T2WUFC0	Wake-Up Field C, LSB	RW	0x2FFF
0x4000251C	T2WUFC1	Wake-Up Field C, MSB	RW	0x0000
0x40002520	T2WUFD0	Wake-Up Field D, LSB	RW	0x3FFF
0x40002524	T2WUFD1	Wake-Up Field D, MSB	RW	0x0000
0x40002528	T2IEN	Interrupt enable	RW	0x0000
0x4000252C	T2STA	Status	R	0x0000
0x40002530	T2CLRI	Clear interrupts	W	Not applicable
0x40002534	RESERVED	Reserved		
0x40002538	RESERVED	Reserved		
0x4000253C	T2WUFA0	Wake-Up Field A, LSB	R	0x1900
0x40002540	T2WUFA1	Wake-Up Field A, MSB	R	0x0000

Current Count Value—LS Halfword Register

Address: 0x40002500, Reset: 0x0000, Name: T2VAL0

Table 68. Bit Descriptions for T2VAL0

Bits	Bit Name	Description	Reset	Access
[15:0]	T2VALL	Current count low. Least significant halfword of current count value.	0x0	R

Current Count Value—MS Halfword Register

Address: 0x40002504, Reset: 0x0000, Name: T2VAL1

Table 69. Bit Descriptions for T2VAL1

Bits	Bit Name	Description	Reset	Access
[15:0]	T2VALH	Current count high. Most significant halfword of current count value.	0x0	R

Control Register

Address: 0x40002508, Reset: 0x0040, Name: T2CON

Table 70. Bit Descriptions for T2CON

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
11	STOP_WUFA	Disables updating Field A Register T2WUFA. This bit, when set, stops the Wake-Up Field A Register T2WUFA from getting updated with the interval register I2INC value. This allows the user to update the interval T2INC or T2WUFA registers safely.	0x0	RW
[10:9]	CLK	Clock select. 00: PCLK (default). 01: LFXTAL. 10: LFOSC. 11: reserved.	0x0	RW
8	WUEN	Wake up enable. Set by the user to enable the wake up request, which wakes up the system because the core clock is off when in power down modes. Cleared by the user to disable the wake-up request.	0x0	RW
7	ENABLE	Timer enable. This bit is set by user code to enable the timer. This bit is cleared by user code to disable the timer (default).	0x0	RW
6	MOD	Timer mode. Set by the user to operate in free running mode (default). Cleared by the user to operate in periodic mode. In this mode, the timer counts up to T2WUFD.	0x1	RW
[5:4]	RESERVED	Reserved. These bits are written 0.	0x0	RW
3	FREEZE	Freeze enable. Set by user to enable the freeze of the high 16 bits after the lower bits have been read from T2VAL0. This ensures that the software reads an atomic shot of the timer. T2VAL1 unfreezes after it has been read. Cleared by the user to disable this feature (default).	0x0	RW
2	RESERVED	Reserved.	0x0	RW
[1:0]	PRE	Prescaler. 00: source clock/1 (default). If the selected clock source is core clock, this setting results in a prescaler of 4. 01: source clock/16. 10: source clock/256. 11: source clock/32,768.	0x0	RW

12-Bit Interval for Wake-Up Field A Register

Address: 0x4000250C, Reset: 0x00C8, Name: T2INC

Table 71. Bit Descriptions for T2INC

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
[11:0]	INTERVAL	Interval for Wake-Up Field A.	0x0C8	RW

Wake-Up Field B—LS Halfword Register

Address: 0x40002510, Reset: 0x1FFF, Name: T2WUFB0

Table 72. Bit Descriptions for T2WUFB0

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFBL	Wake-Up Field B low. Least significant halfword of Wake-Up Field B.	0x1FFF	RW

Wake-Up Field B—MS Halfword Register

Address: 0x40002514, Reset: 0x0000, Name: T2WUFB1

Table 73. Bit Descriptions for T2WUFB1

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFBH	Wake-Up Field B high. Most significant halfword of Wake-Up Field B.	0x0	RW

Wake-Up Field C—LS Halfword Register

Address: 0x40002518, Reset: 0x2FFF, Name: T2WUFC0

Table 74. Bit Descriptions for T2WUFC0

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFCL	Wake-Up Field C low. Least significant halfword of Wake-Up Field C.	0x2FFF	RW

Wake-Up Field C —MS Halfword Register

Address: 0x4000251C, Reset: 0x0000, Name: T2WUFC1

Table 75. Bit Descriptions for T2WUFC1

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFCH	Wake-Up Field C high. Most significant halfword of Wake-Up Field C.	0x0	RW

Wake-Up Field D—LS Halfword Register

Address: 0x40002520, Reset: 0x3FFF, Name: T2WUFD0

Table 76. Bit Descriptions for T2WUFD0

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFD0	Wake-Up Field D low. Least significant halfword of Wake-Up Field D.	0x3FFF	RW

Wake-Up Field D—MS Halfword Register

Address: 0x40002524, Reset: 0x0000, Name: T2WUFD1

Table 77. Bit Descriptions for T2WUFD1

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFDH	Wake-Up Field D high. Most significant halfword of Wake-Up Field D.	0x0	RW

Interrupt Enable Register

Address: 0x40002528, Reset: 0x0000, Name: T2IEN

Table 78. Bit Descriptions for T2IEN

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED		0x0	R
4	ROLL	Rollover interrupt enable. Used only in free running mode. This bit is set by the user to generate an interrupt when Timer 2 rolls over. This bit is cleared by the user to disable the roll over interrupt (default).	0x0	RW
3	WUFD	T2WUFD interrupt enable. This bit is set by user code to generate an interrupt when T2VAL reaches T2WUFD. This bit is cleared by user code to disable T2WUFD interrupt (default).	0x0	RW
2	WUFC	T2WUFC interrupt enable. This bit is set by user code to generate an interrupt when T2VAL reaches T2WUFC. This bit is cleared by user code to disable T2WUFC interrupt (default).	0x0	RW
1	WUFB	T2WUFB interrupt enable. This bit is set by user code to generate an interrupt when T2VAL reaches T2WUFB. This bit is cleared by user code to disable T2WUFB interrupt (default).	0x0	RW
0	WUFA	T2WUFA interrupt enable. This bit is set by user code to generate an interrupt when T2VAL reaches T2WUFA. This bit is cleared by user code to disable T2WUFA interrupt (default).	0x0	RW

Status Register

Address: 0x4000252C, Reset: 0x0000, Name: T2STA

Table 79. Bit Descriptions for T2STA

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	PDOK	Enable bit synchronized. Indicates when a change in the enable bit is synchronized to the 32 kHz clock domain. It is set high when the ENABLE bit in the CONTROL register is set or cleared. It returns low when the change in the ENABLE bit has been synchronized to the 32 kHz clock domain.	0x0	R
7	FREEZE	Timer value freeze. Set automatically to indicate that the value in T2VAL1 is frozen. Cleared automatically when T2VAL1 is read.	0x0	R
6	IRQCRY	Wake-up status to power-down. Set automatically when any of the interrupts are still set in the external crystal clock domain. Cleared automatically when the interrupts are cleared, allowing power-down mode. User code waits for this bit to be cleared before entering power-down mode.	0x0	R
5	RESERVED	Reserved.	0x0	R
4	ROLL	Rollover interrupt flag. Used only in free running mode. Set automatically to indicate a rollover interrupt has occurred. Cleared automatically after a write to T2CLRI.	0x0	R
3	WUFD	T2WUFD interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T2CLRI.	0x0	R
2	WUFC	T2WUFC interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T2CLRI.	0x0	R
1	WUFB	T2WUFB interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T2CLRI.	0x0	R
0	WUFA	T2WUFA interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T2CLRI.	0x0	R

Clear Interrupts Register

Address: 0x40002530, Reset: 0x0000, Name: T2CLRI

Table 80. Bit Descriptions for T2CLRI

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	R
4	ROLL	Rollover interrupt clear. Used only in free running mode. Set by user code to clear a rollover interrupt flag. Cleared automatically after synchronization.	0x0	RW
3	WUFD	T2WUFD interrupt clear.	0x0	RW
2	WUFC	T2WUFC interrupt clear. Set by user code to clear a T2WUFC interrupt flag. Cleared automatically after synchronization.	0x0	RW
1	WUFB	T2WUFB interrupt clear. Set by user code to clear a T2WUFB interrupt flag. Cleared automatically after synchronization.	0x0	RW
0	WUFA	T2WUFA interrupt clear. Set by user code to clear a T2WUFA interrupt flag. Cleared automatically after synchronization.	0x0	RW

Wake-Up Field A—LS Halfword Register

Address: 0x4000253C, Reset: 0x1900, Name: T2WUFA0

Table 81. Bit Descriptions for T2WUFA0

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFAL	Wake-Up Field A low. Least significant halfword of Wake-Up Field A.	0x1900	RW

Wakeup Field A—MS Halfword Register

Address: 0x40002540, Reset: 0x0000, Name: T2WUFA1

Table 82. Bit Descriptions for T2WUFA1

Bits	Bit Name	Description	Reset	Access
[15:0]	T2WUFAH	Wake-Up Field A High. Most significant halfword of Wake-Up Field A.	0x0	RW

WATCHDOG TIMER (WDT)

FEATURES

The WDT is used to recover from an illegal software state. After the WUT is enabled by user code, it requires periodic servicing to prevent it from forcing a reset or interrupt of the processor.

The watchdog timer is clocked either by the 32 kHz crystal oscillator (LFXTAL) or by the 32 kHz on-chip oscillator (LFOSC) (see the CLKCON0: LFCLKMUX register bit in the System Clocks section). It is clocked at all times, except during reset, while in debug mode, and when it is selectively disabled while in hibernate mode.

The watchdog timer is a 16-bit count-down timer with a programmable prescaler. The prescaler source is selectable and can be scaled by a factor of 1, 16, 256, or 4096.

A WDT timeout can generate a reset or an interrupt. The T3CON[1] bit is added to allow selecting of an IRQ instead of a reset, which can be used for debug. The interrupt can be cleared by writing 0xCCCC to the T3CLRI write only register.

BLOCK DIAGRAM

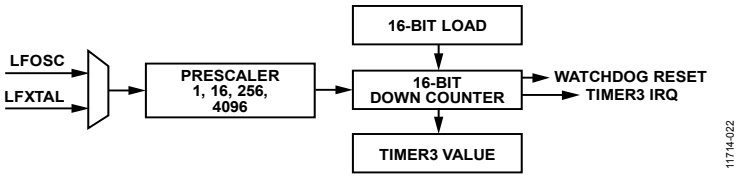


Figure 23. Watchdog Timer Block Diagram

OPERATION

After a valid reset, the watchdog timer is initialized in the hardware as follows:

- T3CON = 0x00E9
- T3LD = 0x1000
- T3VAL = 0x1000

This enables the watchdog timer with a timeout value of 32 seconds. This initial configuration can be modified by user code; however, setting T3CON[5] write protects T3CON and T3LD. Only a reset (depending on T3RCR[0] setting) clears the write protection and allows reconfiguring of the timer. The watchdog timer can be reconfigured at any time when not enabled; that is, when T3CON[5] is not set, user code can change watchdog timer settings at any time.

If the watchdog timer is set to free running mode (T3CON[6] = 0), the watchdog timer value decrements from 0x1000 to 0, wrap around to 0x1000, and continue to decrement. To achieve a timeout value greater than or less than 0x1000 (~32 sec with default prescale = 2), use periodic mode (T3CON[6] = 1), and T3LD and T3CON[3:2] (prescale) written with the values corresponding to the desired timeout period. The maximum timeout is ~8192 sec (T3LD = 0xFFFF, prescale = 3).

At any time, T3VAL contains a valid value to be read, synchronized to the APB clock.

When the watchdog timer decrements to 0, a reset or interrupt is generated. This reset can be prevented by writing T3CLRI with 0xCCCC before the expiration period. A write to T3CLRI causes the watchdog timer to reload with the T3LD (or 0x1000 if in free running mode); the WDT immediately begins a new timeout period and starts to count again. If any value other than 0xCCCC is written, a reset is generated (or an interrupt is generated if selected by T3CON[1]).

While in hibernate mode, the watchdog has the option to suspend operation by setting the PMD bit (T3CON[0]), which is desirable if prolonged periods of time are spent in hibernate mode.

The interrupt output of the watchdog timer needs PCLK active running; that is, it can work in active and CORE_SLEEP modes, but not in SYS_SLEEP or hibernate modes.

The reset output of the watchdog timer works solely from the 32 kHz clock and does not require the system clock to be active; therefore, it can work with all the power-down modes, including hibernate mode.

Watchdog Synchronization

The watchdog timer has three status bits in the T3STA register: CON, LD, and CLRI. These bits indicate that synchronization between fast clock and slow clock domains is in progress for T3CON, T3LD, and T3CLRI registers, respectively. Do not write to the T3CON, T3LD, and T3CLRI registers while a corresponding synchronization bit is asserted.

Example Use Case

The user keeps the default 256 prescale but selects a load value of 0x70 to achieve a ~1 sec type timeout. This results in a count of 256 × 0x70 = 28,672 cycles at 32 kHz.

Factors to consider in this example include the following:

- The user allocates watchdog writes in their code, but there is no timer assisting with when to kick the dog. In such a scenario, two clear watchdog write events may occur in succession.
- Just after a clear watchdog write (T3CLRI), the user cannot write again for 3 × 32 kHz cycles. This represents the first 0.01% of the watchdog period (3/28,672). There is still 99.99% of the timeout period remaining.

In this example, the recommended user sequence at the time the watchdog servicing routine is entered is as follows:

1. Read the status register (T3STA). If CLRI is set, indicating the watchdog was recently serviced, exit (user has 99.99% of the cycle remaining).
2. If CLRI is clear, write 0xCCCC to the clear interrupt register (T3CLRI). This reloads the watchdog counter.

Watchdog Synchronization and Hibernate Entry

The watchdog timer has the option to remain counting while in hibernate mode (PMD bit in T3CON). In such cases, it is advisable to clear the watchdog before hibernate mode is entered. Because the synchronization process requires 3 × 32 kHz cycles, the user must use caution and not enter hibernate mode before the synchronization process is complete. Therefore, the user must monitor the CLRI bit in T3STA, ensuring it is clear, before entering hibernate mode.

WDT MEMORY MAPPED REGISTERS

WDT Register Map

Table 83. Watchdog Timer Register Summary

Address	Name	Description	Reset	RW
0x40002580	T3LD	Load value	0x1000	RW
0x40002584	T3VAL	Current count value	0x1000	R
0x40002588	T3CON	Control	0x00E9	RW
0x4000258C	T3CLRI	Clear interrupt	0x0000	W
0x40002590	T3RCR	Reset control register	0x0000	RW
0x40002594	T3VALA	Value register, asynchronous	0x1000	R
0x40002598	T3STA	Status	0x0000	R

Load Value Register

Address: 0x40002580, Reset: 0x1000, Name: T3LD

Table 84. Bit Descriptions for T3LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value.	0x1000	RW

Current Count Value Register

Address: 0x40002584, Reset: 0x1000, Name: T3VAL

Table 85. Bit Descriptions for T3VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	CCOUNT	Current count value. This register is synchronized to the APB clock. Read only register.	0x1000	R

Control Register

Address: 0x40002588, Reset: 0x00E9, Name: T3CON

Table 86. Bit Descriptions for T3CON

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x1	R
6	MOD	Timer mode. Set by the user to operate in periodic mode (default). Cleared by the user to operate in free running mode. Note that in free running mode, it wraps around at 0x1000.	0x1	RW
5	ENABLE	Timer enable. Set by the user to enable the timer (default). Cleared by the user to disable the timer.	0x1	RW
4	RESERVED	Reserved.	0x0	R
[3:2]	PRE	Prescaler. 00: source clock/1. If the selected clock source is the core clock, then this setting results in a prescaler of 4. 01: source clock/16. 10: source clock/256 (default). 11: source clock/4096.	0x2	RW
1	IRQ	Timer interrupt. Set by the user to generate an interrupt when the timer times out. This feature is provided for debug purposes and is only available in active mode. Cleared by the user to generate a reset on a time out (default).	0x0	RW
0	PMD	Power mode disable. PMD controls the behavior of the watchdog when in hibernate mode or debug mode. If the application requires prolonged periods of time spent in hibernate mode and it is not desirable to periodically wake up to service the watchdog timer, the counter within the watchdog timer can be suspended when entering the hibernate power mode. Regardless of how the PMD bit is set, it is recommended that the watchdog timer be cleared before entering hibernate mode. 0: the watchdog timer continues its count down while in hibernate mode or debug mode. 1: when hibernate mode or debug mode is entered, the watchdog counter suspends its count down. As hibernate mode is exited, the countdown resumes from its current count value (the count is not reset).	0x1	RW

Clear Interrupt Register

Address: 0x4000258C, Reset: 0x0000, Name: T3CLRI

Table 87. Bit Descriptions for T3CLRI

Bits	Bit Name	Description	Reset	Access
[15:0]	CLRWDG	Clear watchdog. User writes 0xCCCC to reset/reload/restart T3 or clear IRQ. A write of any other value causes a watchdog reset/IRQ. Write only, reads 0.	0x0	W

Reset Control Register

Address: 0x40002590, Reset: 0x0000, Name: T3RCR

Table 88. Bit Descriptions for T3RCR

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0	R
0	RESET_CTRL	WDT reset configuration bit. 0: POR or system reset resets WDT. 1: only POR resets WDT.	0x0	RW

Value Register Asynchronous

Address: 0x40002594, Reset: 0x1000, Name: T3VALA

Table 89. Bit Descriptions for T3VALA

Bits	Bit Name	Description	Reset	Access
[15:0]	VALA	Current WDT count value. Unsynchronized to APB clock.	0x1000	R

Status Register

Address: 0x40002598, Reset: 0x0000, Name: T3STA

Table 90. Bit Descriptions for T3STA

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	R
4	LOCK	Lock status bit. Set automatically in hardware if T3CON[5] has been set by user code. Cleared by default and until user code sets T3CON[5].	0x0	R
3	CON	T3CON write sync in progress. 0: APB and T3 clock domains T3CON configuration values match. 1: APB T3CON register values are being synchronized to T3 clock domain.	0x0	R
2	LD	T3LD write sync in progress. 0: APB and T3 clock domains T3LD values match. 1: APB T3LD value is being synchronized to T3 clock domain.	0x0	R
1	CLRI	T3CLRI write sync in progress. 0: APB T3CLRI write sync not done. 1: APB T3CLRI write is being synced to T3 clock domain. T3 is restarted (if 0xCCCC was written) after sync is complete.	0x0	R
0	IRQ	WDT interrupt. 0: T3 interrupt not pending. 1: T3 interrupt pending.	0x0	R

BUS MATRIX

FEATURES

The AMBA system bus matrix of the ADuCM350 provides the communication fabric between the ARM Cortex-M3 processor, the AFE, the system memory, and the peripherals.

As a multilayer system bus, it allows up to five masters to be active on the bus at any one time. Each slave has configurable priorities to decide which master has higher priority than the others.

The five AHB master interfaces include the following:

- The Cortex-M3 data bus (DBus)
- The Cortex-M3 system bus (SBus)
- The DMA Bus 0
- The DMA Bus 1
- The USB DMA

There are eight slave interfaces implemented. These include the following:

- Flash memory
- SRAM 0
- SRAM 1
- AFE
- USB controller
- 32-bit APB
- 16-bit APB without DMA access
- 16-bit APB with DMA access

Additional AHB features that are provided in the AMBA bus matrix include the following:

- Reconfigurable access priority at each slave interface.
- Automatic enabling of peripheral clocks if the APB is accessed.
- Support for 2× USB master and slave interface frequency.
- ×1 frequency of 32-bit APB interface.
- An equal or integer divided frequency of the 16-bit APB interface.
- A 3-word depth, FIFO-based, AHB to APB bridge for the 16-bit APB interface.

ARCHITECTURE

Figure 24 shows the key functionality and connectivity of the bus matrix. Three clock domains are used: the PCLK for the 16-bit APB interfaces, the USBCTLCLK for the USB system interface, and the HCLK clocks everything else, including the 32-bit APB interface.

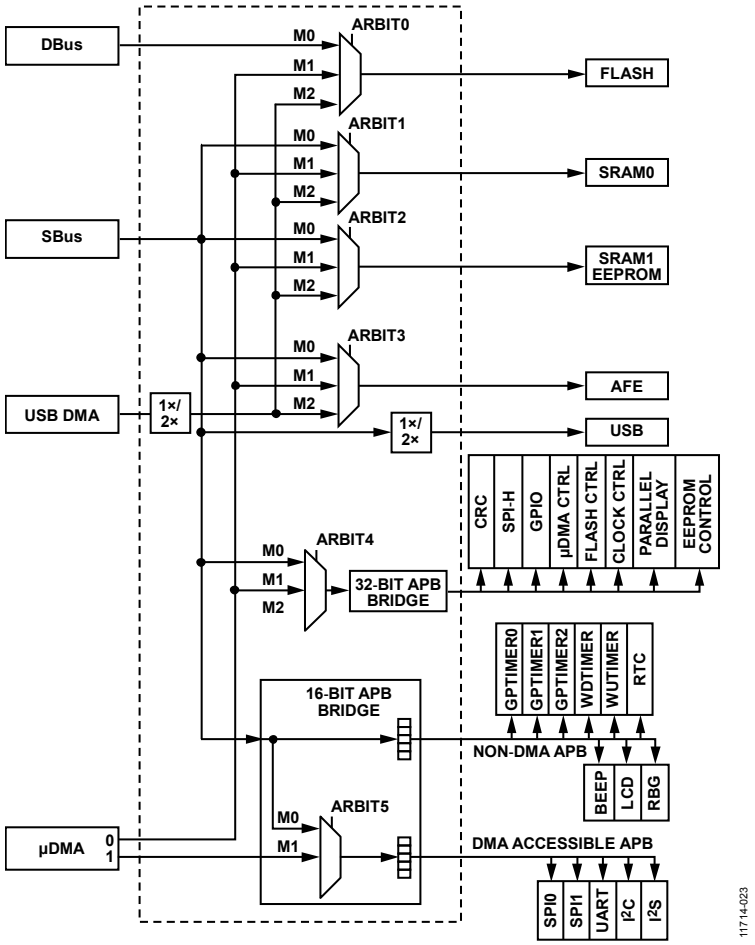


Figure 24. AMBA Bus Matrix Block Diagram

OPERATION

The bus matrix contains the ARBIT0 to ARBIT5 registers to change the default and the priority for each slave and to set the priority for each of its three masters.

To reduce dynamic power, the peripheral clocks (including UCLK_SPI0, UCLK_SPI1, UCLK_SPIH, UCLK_I2C, UCLK_UART, and UCLK_I2S) can be stopped using the control bits located in the CLKCON5 register. The 16-bit AHB to APB bridge automatically enables the corresponding peripheral clock if there is APB access to that peripheral.

For correct USB operation, the interfaces of that controller must run at 2× the HCLK frequency (32 MHz). This is configured by the CLKCON1 register and the corresponding SPLL at the CLKCON3 register.

The PCLK of the 16-bit APB interface can be equal to or an integral divided frequency of HCLK; that is, these APB slaves can run at much slower frequency than HCLK to save power.

BUS MATRIX MEMORY MAPPED REGISTERS

Bus Matrix Register Map

Table 91. Bus Matrix Register Summary

Address	Name	Description	Reset	RW
0x40028040	ARBIT0	Arbitration Priority Configuration for Slave 0 (Flash).	0x0000	RW
0x40028044	ARBIT1	Arbitration Priority Configuration for Slave 1 (SRAM0).	0x0000	RW
0x40028048	ARBIT2	Arbitration Priority Configuration for Slave 2 (SRAM1 + EEPROM).	0x0000	RW
0x4002804C	ARBIT3	Arbitration Priority Configuration for Slave 3 (AFE).	0x0000	RW
0x40028050	ARBIT4	Arbitration Priority Configuration for Slave 4 (32-bit APB).	0x0000	RW
0x40028054	ARBIT5	Arbitration Priority Configuration for Slave 5 (16-bit APB).	0x0000	RW
0x40028058	RESERVED	Reserved.		
0x4002805C	RESERVED	Reserved.		
0x40028060	RESERVED	Reserved.		
0x40028064	RESERVED	Reserved.		

Arbitration Priority Configuration for Slave 0 (Flash) Register

Address: 0x40028040, Reset: 0x0000, Name: ARBIT0

Table 92. Bit Descriptions for BMARBIT0

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:8]	DEFAULT_ACTIVE0	Default active master when no request. 00: previous active master as the default. 01: Master 0 as default. 10: Master 1 as default. 11: Master 2 as default.	0x0	RW
[7:3]	RESERVED	Reserved.	0x0	R
[2:0]	ARBIT_PRIO_0	Arbiter priority for Slave 0. 000: M0 > M1 > M2. 001: M0 > M2 > M1. 010: M1 > M0 > M2. 011: M1 > M2 > M0. 100: M2 > M1 > M0. 101: M2 > M0 > M1. 110: reserved. 111: reserved.	0x0	RW

Arbitration Priority Configuration for Slave 1 (SRAM0) Register

Address: 0x40028044, Reset: 0x0000, Name: ARBIT1

Table 93. Bit Descriptions for BMARBIT1

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:8]	DEFAULT_ACTIVE1	Default active master when no request. 00: previous active master as the default. 01: master 0 as default. 10: master 1 as default. 11: master 2 as default.	0x0	RW
[7:3]	RESERVED	Reserved.	0x0	R

Bits	Bit Name	Description	Reset	Access
[2:0]	ARBIT_PRIO_1	Arbiter priority for Slave 1. 000: M0 > M1 > M2. 001: M0 > M2 > M1. 010: M1 > M0 > M2. 011: M1 > M2 > M0. 100: M2 > M1 > M0. 101: M2 > M0 > M1. 110: reserved. 111: reserved.	0x0	RW

Arbitration Priority Configuration for Slave 2 (SRAM1 + EEPROM) Register

Address: 0x40028048, Reset: 0x0000, Name: ARBIT2

Table 94. Bit Descriptions for BMARBIT2

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:8]	DEFAULT_ACTIVE2	Default active master when no request. 00: previous active master as the default. 01: Master 0 as default. 10: Master 1 as default. 11: Master 2 as default.	0x0	RW
[7:3]	RESERVED	Reserved.	0x0	R
[2:0]	ARBIT_PRIO_2	Arbiter priority for Slave 2. 000: M0 > M1 > M2. 001: M0 > M2 > M1. 010: M1 > M0 > M2. 011: M1 > M2 > M0. 100: M2 > M1 > M0. 101: M2 > M0 > M1. 110: reserved. 111: reserved.	0x0	RW

Arbitration Priority Configuration for Slave 3 (AFE) Register

Address: 0x4002804C, Reset: 0x0000, Name: ARBIT3

Table 95. Bit Descriptions for BMARBIT3

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:8]	DEFAULT_ACTIVE3	Default active master when no request. 00: previous active master as the default. 01: Master 0 as default. 10: Master 1 as default. 11: Master 2 as default.	0x0	RW
[7:3]	RESERVED	Reserved.	0x0	R
[2:0]	ARBIT_PRIO_3	Arbiter priority for Slave 3. 000: M0 > M1 > M2. 001: M0 > M2 > M1. 010: M1 > M0 > M2. 011: M1 > M2 > M0. 100: M2 > M1 > M0. 101: M2 > M0 > M1. 110: reserved. 111: reserved.	0x0	RW

Arbitration Priority Configuration for Slave 4 (32-Bit APB) Register

Address: 0x40028050, Reset: 0x0000, Name: ARBIT4

Table 96. Bit Descriptions for BMARBIT4

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:8]	DEFAULT_ACTIVE4	Default active master when no request. 00: previous active master as the default. 01: Master 0 as default. 10: Master 1 as default. 11: Master 2 as default.	0x0	RW
[7:3]	RESERVED	Reserved.	0x0	R
[2:0]	ARBIT_PRIO_4	Arbiter priority for Slave 4. 000: M0 > M1 > M2. 001: M0 > M2 > M1. 010: M1 > M0 > M2. 011: M1 > M2 > M0. 100: M2 > M1 > M0. 101: M2 > M0 > M1. 110: reserved. 111: reserved.	0x0	RW

Arbitration Priority Configuration for Slave 5 (16-Bit APB with DMA access) Register

Address: 0x40028054, Reset: 0x0000, Name: ARBIT5

Table 97. Bit Descriptions for BMARBIT5

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:8]	DEFAULT_ACTIVE5	Default active master when no request. 00: previous active master as the default. 01: Master 0 as default. 10: Master 1 as default. 11: Master 2 as default.	0x0	RW
[7:3]	RESERVED	Reserved.	0x0	R
[2:0]	ARBIT_PRIO_5	Arbiter priority for Slave 5. 000: M0 > M1 > M2. 001: M0 > M2 > M1. 010: M1 > M0 > M2. 011: M1 > M2 > M0. 100: M2 > M1 > M0. 101: M2 > M0 > M1. 110: reserved. 111: reserved.	0x0	RW

POWER MANAGEMENT UNIT

INTRODUCTION

The power management unit (PMU) provides control of the ADuCM350 power modes and allows the ARM Cortex-M3 processor to control the clocks and power gating to reduce the dynamic and standby power.

The PMU is in the always on (always powered) section of the ADuCM350. Four power modes are available; each mode provides an additional low power benefit with a corresponding reduction in functionality. In hibernate mode, the ADuCM350 consumes 2 µA typical with the RTC enabled. In the fully operational active mode, the ADuCM350 consumes ~300 µA/MHz, depending on what features are being used and the application running on the processor. Table 98 lists all of the available power modes.

Note that backup mode, discussed in the Power Supplies section, is independent of the PMU and can overlap with any of the four power modes.

Features of the power management unit include the following:

- Four distinct power modes
- Clock gating to reduce the dynamic power consumption
- Power gating to reduce the static power consumption in hibernate mode
- State retention during hibernate mode
- Wake-up time of 10 µs from hibernate mode
- Automatic clock gating of various modules when no accesses are made to them
- Multiple wake-up sources available

Table 98. Summary of Power Modes

Mode	Description	Wake-Up Time	Wake-Up Source
0 (Active)	All modules are on. Some modules can be turned on or off by user-programmable registers. The system enters this mode upon power-up and after waking up from sleep.	Not available	Not available
1 (CORE_SLEEP)	Cortex-M3 core is in sleep[1] mode and its clock is gated. Peripherals can be turned on or off via gating by the user. Power comes from HP LDO.	Immediate (~1 clock cycle of system clock)	All wake-up sources
2 (SYS_SLEEP)	Cortex-M3 core is in sleep mode and all peripherals are clock gated. Power comes from HP LDO. Analog clocks can still be enabled by the user.	Immediate (~1 clock cycle of system clock)	System reset, NMI, SYSTICK, WDT, WUT, USB, RTC, CapTouch, GPIO
3 (Hibernate)	Cortex-M3 is in deep sleep[2] and power gating is applied to the power gated area with state retention. LP LDO provides power to the always on section of the ADuCM350.	10 µs[3]	System reset, WDT, WUT, USB, RTC, GPIO, NMI

[1] In Cortex-M3 sleep mode, Cortex-M3 core clock (HCLK_CORE) is gated.
[2] In Cortex-M3 deep sleep mode, Cortex-M3 core clock (HCLK_CORE) and interrupt controller clock (FCLK) are gated.
[3] This is a minimum time assuming wake up with the internal oscillator. If using a crystal or a PLL, its wake-up time must be considered.

POWER MODES

PM0: Active

The system is fully active and the processor is executing instructions. The user can choose to disable clocks to modules that are not being used (see Table 99). Memories are clocked only when they are used. Note that the Cortex-M3 manages its internal clocks and can be in a partial clock gated state even in active mode.

Automatic clock gating is used on some modules and is transparent to the user. User code can call the wait for interrupt (WFI) instruction to put the Cortex-M3 in sleep mode and enter the power-down state configured by the PWRMOD register. When the ADuCM350 wakes up from any of the low power modes, it returns to active mode.

Figure 25 shows the ADuCM350 in full active mode, where power is supplied by the HP LDO.

Figure 26 shows the same mode with some selected modules clock gated to reduce power.

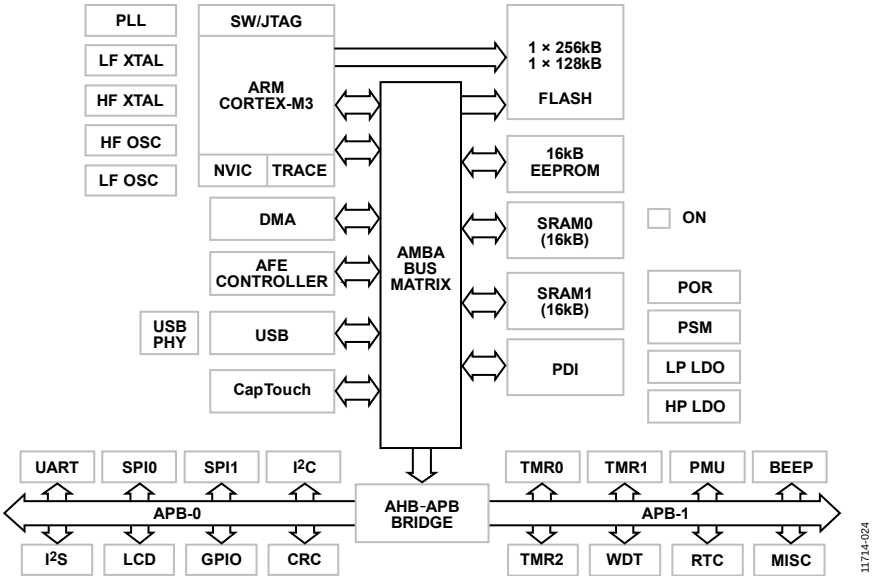


Figure 25. Active Mode

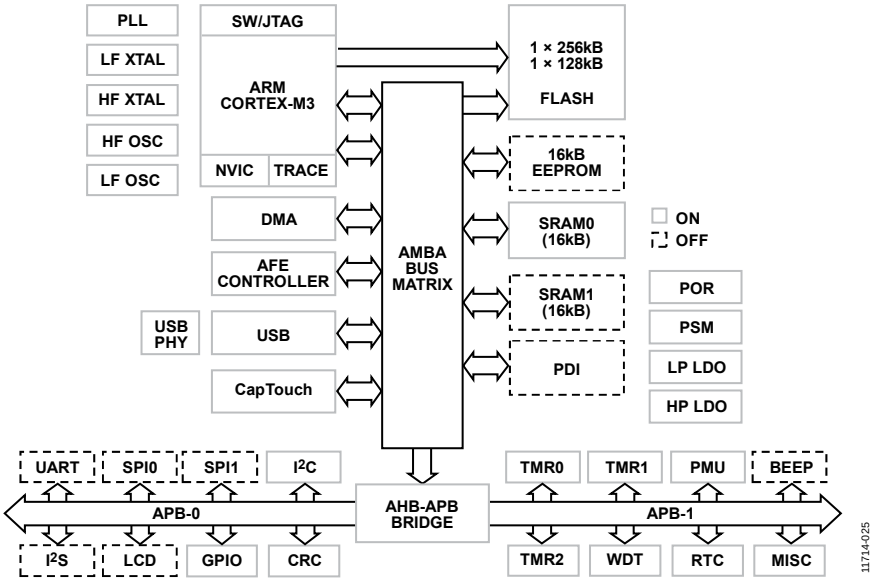


Figure 26. Example of Active Mode with Clock Gating

The following steps are executed to send the system into a given power mode:

- Write the power mode value to the PWRMOD register (Address 0x40002400).
- For hibernate mode, write to the SLEEPDEEP bit of the INTCON0 register (Address 0xE000ED10). As an option, the sleep on exit feature can also be enabled by writing to the SLEEPONEXIT bit of the INTCON0 register (Address 0xE000ED10).

Program the Cortex-M3 to execute a WFI instruction.

PM1: CORE_SLEEP

In CORE_SLEEP mode, the system gates the clock to the Cortex-M3 core after it has entered sleep mode. The rest of the system remains active. No instructions can be executed; however, DMA transfers can continue to occur between peripherals and memories.

This mode has the advantage of eliminating instruction accesses to flash memory as well as usage of stack and temporary variables in SRAM0, significantly reducing dynamic power in the system. DMA accesses into SRAM or flash automatically enables the clocks to that particular block. The NVIC clock (FCLK) remains active, and the NVIC processes wake-up events.

Figure 27 shows the ADuCM350 in the default CORE_SLEEP mode with only the core clock gated. Figure 28 shows an example where various submodules are also clock gated in CORE_SLEEP mode to provide additional power reduction. See Table 99 for more details.

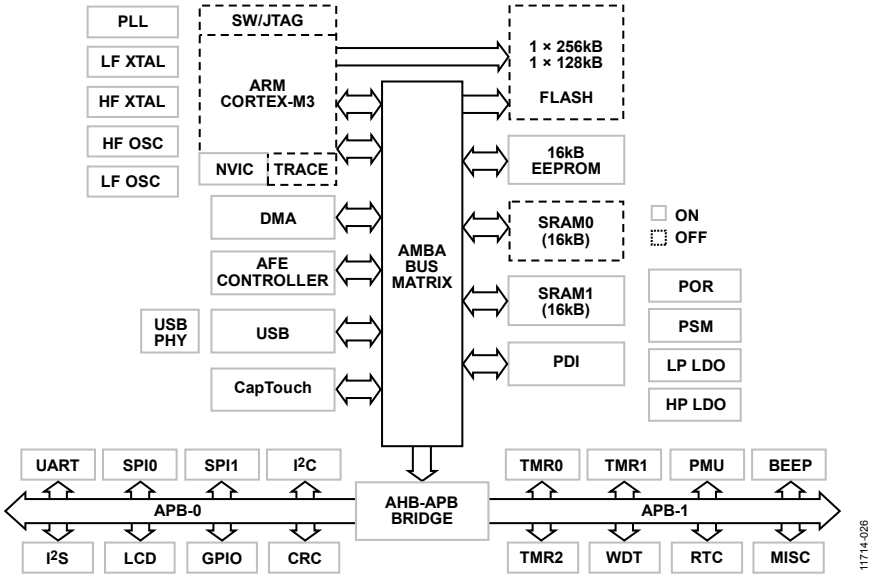


Figure 27. CORE_SLEEP Mode

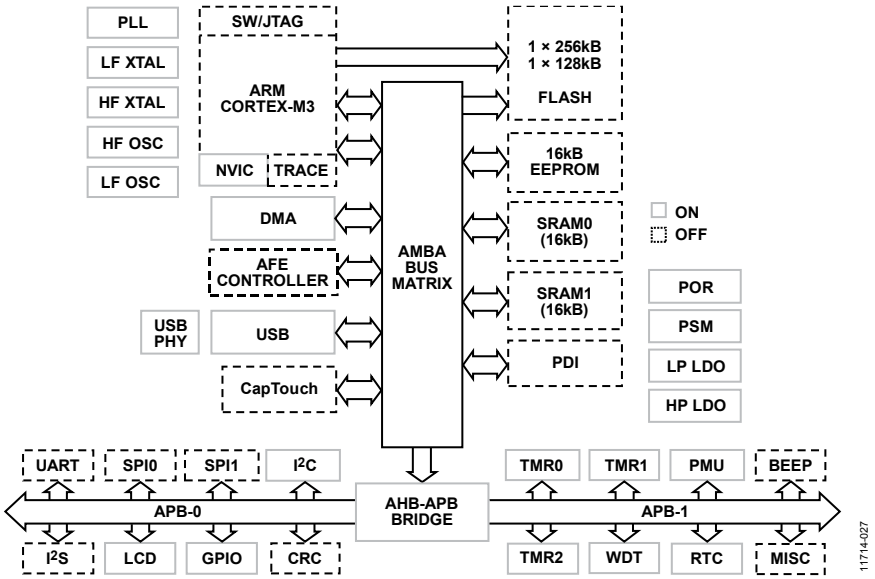


Figure 28. CORE_SLEEP Mode with Some Modules Clock Gated

PM2: SYS_SLEEP

In SYS_SLEEP mode, the system gates HCLK (system bus clock) and PCLK (peripheral bus clock) after the Cortex-M3 has entered sleep mode. The gating of these clocks stops all AHB attached masters/slaves and all peripherals attached to APB. Peripheral clocks are all off, and they are no longer user programmable. The NVIC (interrupt controller) clock (FCLK) remains active, and the NVIC processes wake-up events. See Table 99 for more details.

Figure 29 shows an example use case of SYS_SLEEP mode, where the majority of the ADuCM350 modules are clock gated.

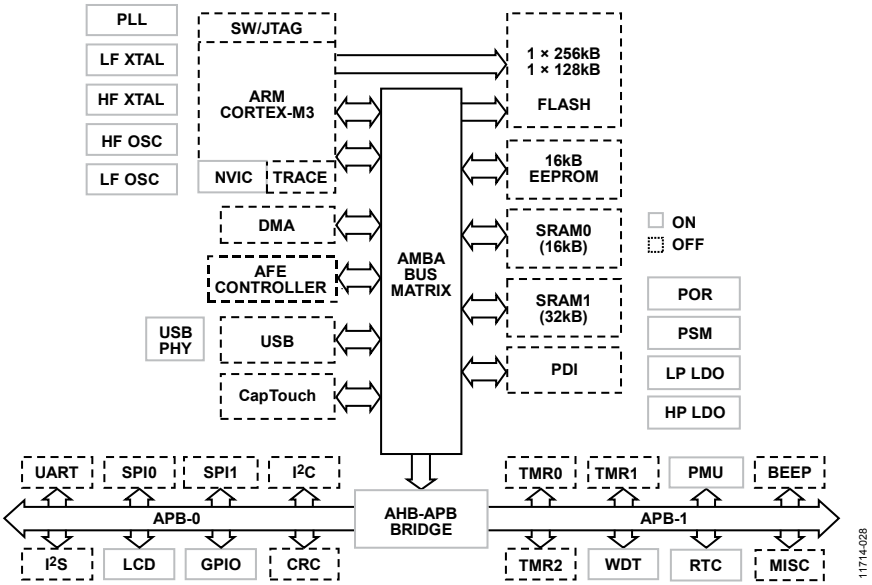


Figure 29. SYS_SLEEP Mode Example Use Case

PM3: Hibernate

In hibernate mode, the system disables power to all combinational logic and places sequential logic in retain mode. Because FCLK is stopped, the number of sources capable of waking up the system is restricted. The sources listed in Table 100 are the only sources able to wake up the system.

In hibernate mode, SRAM0 has the option of retaining its contents (see the Power Gating section). There is no option to retain the contents of SRAM1 while in hibernate mode. The HP LDO is disabled upon entering hibernate mode, and the LP LDO is used to provide DVDD system power. Note that the LP LDO always supplies power to the RTC. The VCCM PSM is software programmable in hibernate mode. Figure 30 shows hibernate mode.

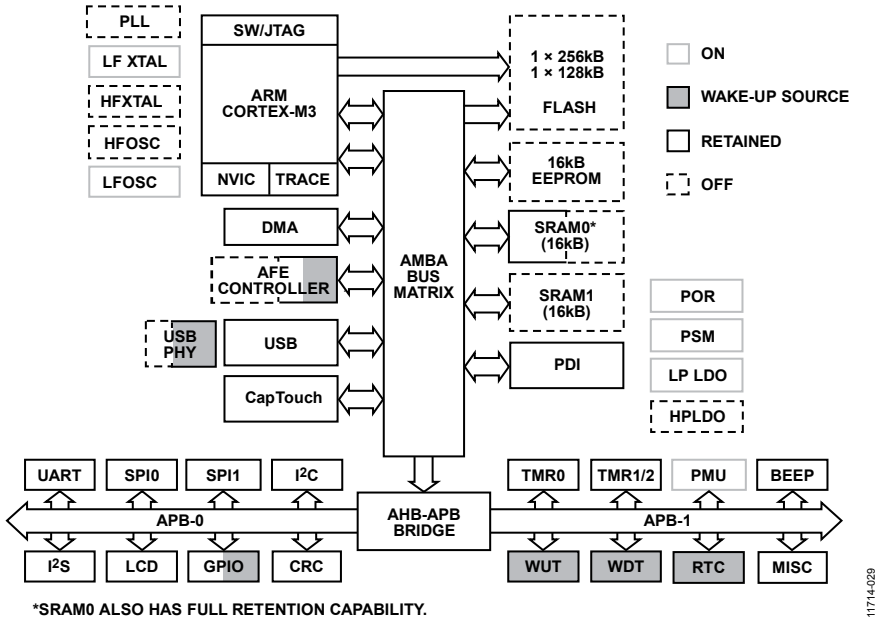


Figure 30. Hibernate Mode

POWER MODE CONTROL TABLE

Table 99 summarizes all power modes with respect to what units are on or off for a given mode. Features that are listed as on/off default to on; however, they can be disabled in software. Features that are listed as off/on default to off; however, they can be enabled by software. Note that the beeper and the LCD use a 32 kHz clock and are, therefore, not explicitly listed in the table.

Table 99. Details of the Power Modes

Block	Active	CORE_SLEEP	SYS_SLEEP	Hibernate
Cortex-M3	On	Sleep	Sleep	Deep sleep
Power Gating	No	No	No	Yes
HP LDO	On	On	On	Off
LP LDO	On	On	On	On
HF XTAL OSC	Off/on	Off/on	Off/on	Off
LF XTAL OSC	Off/on	Off/on	Off/on	Off/on
Lower Half SRAM0 Power	On	On	On	Retain
Higher Half SRAM0 Power	On	On	On	Off/retain
SRAM1 Power	On	On	On	Off
USB RAM Power	On	On	On	Off
HF OSC	On/off	On/off	On/off	Off
LF OSC	On/off	On/off	On/off	On/off
UPLL	Off/on	Off/on	Off/on	Off
SPLL	Off/on	Off/on	Off/on	Off
FCLK (Interrupt)	On	On	On	Off
HCLK_CORE	On	Off	Off	Off
HCLK_BUS (AHB)	On	On	Off	Off
HCLK_CT (CapTouch)	On	On	Off/on	Off

Block	Active	CORE_SLEEP	SYS_SLEEP	Hibernate
PCLK (APB)	On	On	Off	Off
UCLK_SPI0	On/off[1]	On/off[1]	Off	Off
UCLK_SPI1	On/off[1]	On/off[1]	Off	Off
UCLK_SPIH	On/off[1]	On/off[1]	Off	Off
UCLK_I2C	On/off[1]	On/off[1]	Off	Off
UCLK_UART	On/off[1]	On/off[1]	Off	Off
UCLK_I2S	On/off[1]	On/off[1]	Off	Off
USBCTLCLK	On/off[2]	On/off[2]	Off	Off
USBPHYCLK	On/off[2]	On/off[2]	Off	Off
ACLK (AFE)	Off/on	Off/on	Off/on	Off
AFE_ADC_CLK (AFE)	Off/on	Off/on	Off/on	Off
CTCLK (CapTouch)	Off/on	Off/on	Off/on	Off

[1] These peripheral clocks are automatically switched on if the registers of the peripheral are accessed.
[2] USB clocks are automatically turned off if the USB bus is idle for more than 3 ms (suspend), a USB link power management (LPM) command from the host is received and acknowledged, or if disconnection occurs.

WAKE-UP

Wake-Up Sequence

The system wakes up with the same operating state that was present during power down. The modules in operation before power down are enabled automatically at wake up. The system waits on the appropriate ready signals from the clock sources (such as PLL lock and crystal ok). Wake-up time is a function of which blocks must be reenabled during the wake-up sequence. The minimum wake-up time requires the HP LDO and internal oscillator to power up (around 10 µs).

Wake-Up Sources

Table 100 shows the wake-up interrupt sources for the different power modes. The ADuCM350 Notes column suggests some product specific sources for some of these interrupts.

Table 100. Wake-Up Sources

Source	CORE_SLEEP	SYS_SLEEP	Hibernate	ADuCM350 Notes
Real-Time Clock	Yes	Yes	Yes	
CapTouch	Yes	Yes	No	
AFE	Yes	Yes	No	
Wake-Up Timer	Yes	Yes	Yes	
Nonmaskable Interrupts	Yes	Yes	Yes	
External IRQ 0	Yes	Yes	Yes	GPIO P4.0
External IRQ 1	Yes	Yes	Yes	Power button
External IRQ 2	Yes	Yes	Yes	Select button
External IRQ 3	Yes	Yes	Yes	Cursor Button 0
External IRQ 4	Yes	Yes	Yes	Cursor Button 1
External IRQ 5	Yes	Yes	Yes	Cursor Button 2
External IRQ 6	Yes	Yes	Yes	Cursor Button 3
External IRQ 7	Yes	Yes	Yes	GPIO P3.4
External IRQ 8	Yes	Yes	Yes	GPIO P0.10
Watchdog	Yes	Yes	Yes	
Flash Controllers	Yes	No	No	
GP Timers	Yes	No	No	
Peripherals	Yes	No	No	
DMA Error	Yes	No	No	
DMA Done	Yes	No	No	
USB Wake Up	Yes	Yes	Yes	
USB IRQ and DMA	Yes	No	No	
XOSC Event	Yes	Yes	No	
PLL Event	Yes	Yes	No	

Clock Gating

Clock gating is used in active, CORE_SLEEP, and SYS_SLEEP modes to reduce dynamic power. See the general clocking diagram (Figure 4) within the System Clocks section for more information. Individual gates are available for every clock shown. Certain clocks are gated depending on power mode, and others are user programmable. Clocks for many of the modules (for example, memories and DMA) are clock gated automatically when not in use. Table 101 summarizes the clock gating across the system.

Table 101. Clock Gating

Clock/Mode	Active	CORE_SLEEP	SYS_SLEEP
FCLK	On	On	On
HCLK_CORE	On	Off	Off
HCLK_BUS	On	On	Off
HCLK_CT	On	On	User
PCLK	On	On	Off
UCLK_SPI0	User	User	Off
UCLK_SPI1	User	User	Off
UCLK_SPIH	User	User	Off
UCLK_I2C	User	User	Off
UCLK_UART	User	User	Off
UCLK_I2S	User	User	Off
USBPHYCLK	User/automatic	User/automatic	Off
USBCTLCLK	Automatic	Automatic	Off
ACLK	User	User	User
AFE_ADC_CLK	User	User	User
CTCLK	User	User	User

Automatic Clock Gating

Some blocks use a certain level of automatic gating if they are not being accessed to further reduce dynamic power consumption. Table 102 provides a brief overview of this feature.

Table 102. Automatic Clock Gating

Module	Description
Peripherals	In active and CORE_SLEEP modes, the APB peripherals (including I^2C, SPI, and UART) turn themselves on automatically if a bus transfer is detected on their registers.
USB	The USB clocks are turned off automatically if any of the following three events occur: the USB is idle for more than 3 ms, an explicit LPM command is received from the host, or a disconnection occurs.
SRAM	In active and CORE_SLEEP modes, the SRAM clock is automatically gated if no accesses to it are detected.
DMA	In active and CORE_SLEEP mode, the DMA clock is automatically gated if no DMA transfers are performed.
Flash and EEPROM	In active and CORE_SLEEP modes, the clock to the flash memory and EEPROM controllers is stopped when there are no accesses to them via the bus matrix. These clocks are automatically enabled when they are accessed.

POWER GATING

Clock gating provides a means of reducing dynamic power consumption. The ADuCM350 also uses power gating to provide a large reduction in static power (leakage). Power gating is only enabled when the ADuCM350 enters hibernate mode. This mode switches power off to combinatorial logic and places sequential elements into a low leakage retain state. The first 8 kB of system SRAM0 (Byte 0x0000 through Byte 0x1FFF) are always retained in hibernate mode. System SRAM0 Byte 0x2000 through Byte 0x3FFF have the option of being retained in hibernate mode. SRAM1 and USB RAM are never retained in hibernate mode.

Figure 31 shows the power gating of the digital system. During hibernate mode, the LP LDO supplies power for the system in its low power retain state. Portions of RAM0 can be optionally retained by setting the RAM0_RET bit in Register PWRMOD.

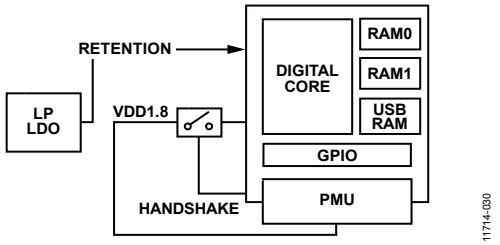


Figure 31. Power Switch

POWER GATED MODULES

The following list indicates which modules in hibernate mode are power gated (sequential elements are retained) and which modules in hibernate mode are not power gated (sequential elements are reset).

Table 103. Power Gated in Hibernate

Module	Power Gated	Comments
Cortex-M3	Yes	
General-Purpose and Instruction Flash Controllers	Yes	
RAM Bank 0	Yes	8 kB and 16 kB gated options
RAM Bank 1	No	RAM contents are lost when hibernate is entered
AMBA Bus Matrix	Yes	
General-Purpose Timers (0, 1, and 2)	Yes	
System Power Control	No	Always powered
External Interrupts	No	Always powered
System Reset	No	Always powered
Wake-Up Timer	No	Always powered
Watchdog Timer	No	Always powered
Real-Time Clock	No	Always powered
I^2C Master/Slave	Yes	
SPI 0-1-H	Yes	
UART	Yes	
I^2S Master/Slave	Yes	
Beeper	Yes	
Random Bit Generator	Yes	
LCD Controller	Yes	
DMA	Yes	
GPIO	Yes	
CRC Engine	Yes	
Parallel Display Interface	Yes	
AFE	No	Upon exiting hibernate mode, the AFE is in its reset state
CapTouch	Yes	
USB Controller	Yes	USB 2 kB endpoint RAM contents are lost when hibernate mode is entered

EXAMPLE USE CASE

The following diagram shows an example sequence to change to the various power modes.

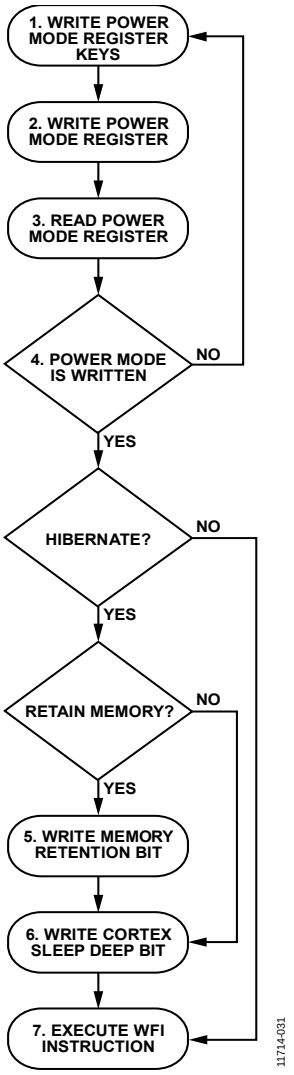


Figure 32. Power Mode Example Sequence

PMU MEMORY MAPPED REGISTERS

PMU Register Map

Table 104. Power Management Register Summary

Address	Name	Description	Reset	RW
0x40002400	PWRMOD	Power modes	0x0000	RW
0x40002404	PWRKEY	Key protection for PWRMOD	0x0000	RW
0xE000ED10	INTCON0	Power modes system control register	0x0000	RW

Power Modes Register

Address: 0x40002400, Reset: 0x0000, Name: PWRMOD

Table 105. Bit Descriptions for PWRMOD

Bits	Bit Name	Description	Reset	Access
15	RAM0_RET	Retention for RAM 0. 0: RAM 0 Bytes 0x2000 to 0x3FFF are not retained during hibernate. 1: RAM 0 Bytes 0x2000 to 0x3FFF are retained during hibernate.	0x0	RW
[14:4]	RESERVED	Reserved. These bits are written 0 by user code.	0x0	R
3	WICENACK	WIC acknowledgment for SLEEPDEEP.	0x0	R
2	RESERVED	Reserved. These bits are written 0 by user code.	0x0	R
[1:0]	PWRMOD	Power modes control bits. When read, these bits contain the last power mode value entered by user code. Note that to place the Cortex-M3 in sleep deep mode for hibernate, the System Control Register of the Cortex-M3 (Address 0xE000ED10) must be configured to 0x4 or 0x06. 00: active mode. 01: CORE_SLEEP mode. 10: SYS_SLEEP mode. 11: hibernate mode.	0x0	RW

Key Protection for PWRMOD Register

Address: 0x40002404, Reset: 0x0000, Name: PWRKEY

Table 106. Bit Descriptions for PWRKEY

Bits	Bit Name	Description	Reset	Access
[15:0]	PWRKEY	Power control key register. The PWRMOD register is key protected. Two writes to the key are necessary to change the value in the PWRMOD register. First 0x4859, then 0xF27B. The PWRMOD register must then be written. A write to any other register on the APB before writing to PWRMOD returns the protection to the lock state.	0x0	RW

System Control Register

Address: 0xE000ED10, Reset: 0x0000, Name: INTCON0

Table 107. Bit Descriptions for INTCON0

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
2	SLEEPDEEP	Deep sleep flag for hibernate mode. 0: sleep deep is not enabled. 1: sleep deep is enabled.	0x0	RW
1	SLEEPONEXIT	Sleeps the core on exit from an ISR. 0: sleep on exit is not enabled. 1: sleep on exit is enabled.	0x0	RW
0	RESERVED	Reserved.	0x0	R

POWER SUPPLIES

INTRODUCTION

In this section, the external power supplies, power supply regulators, and power supply sequencing is outlined. All of the necessary supervisory circuits and voltage regulators are integrated within the ADuCM350. The ADuCM350 is typically powered from a single CR2032 coin cell battery. The battery, which is not rechargeable, has a typical battery discharge profile as shown in Figure 33.

The ADuCM350 USB port can be connected to a peripheral. The VBUS pin on the USB supplies only the circuits associated with the USB interface (PHY). As a result, this regulator is not included in this section. See the Universal Serial Bus Controller section for more details.

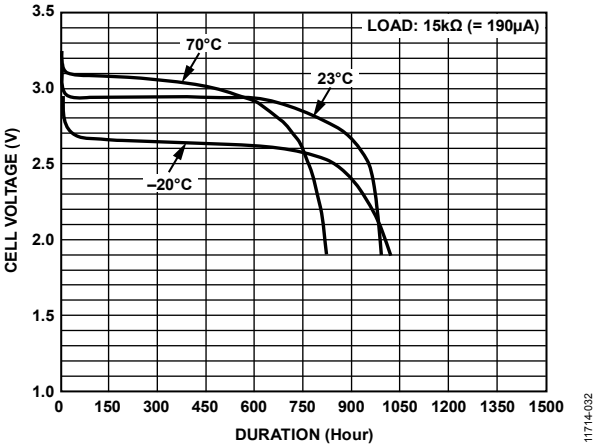


Figure 33. Typical Sanyo CR2032 Discharge Profile

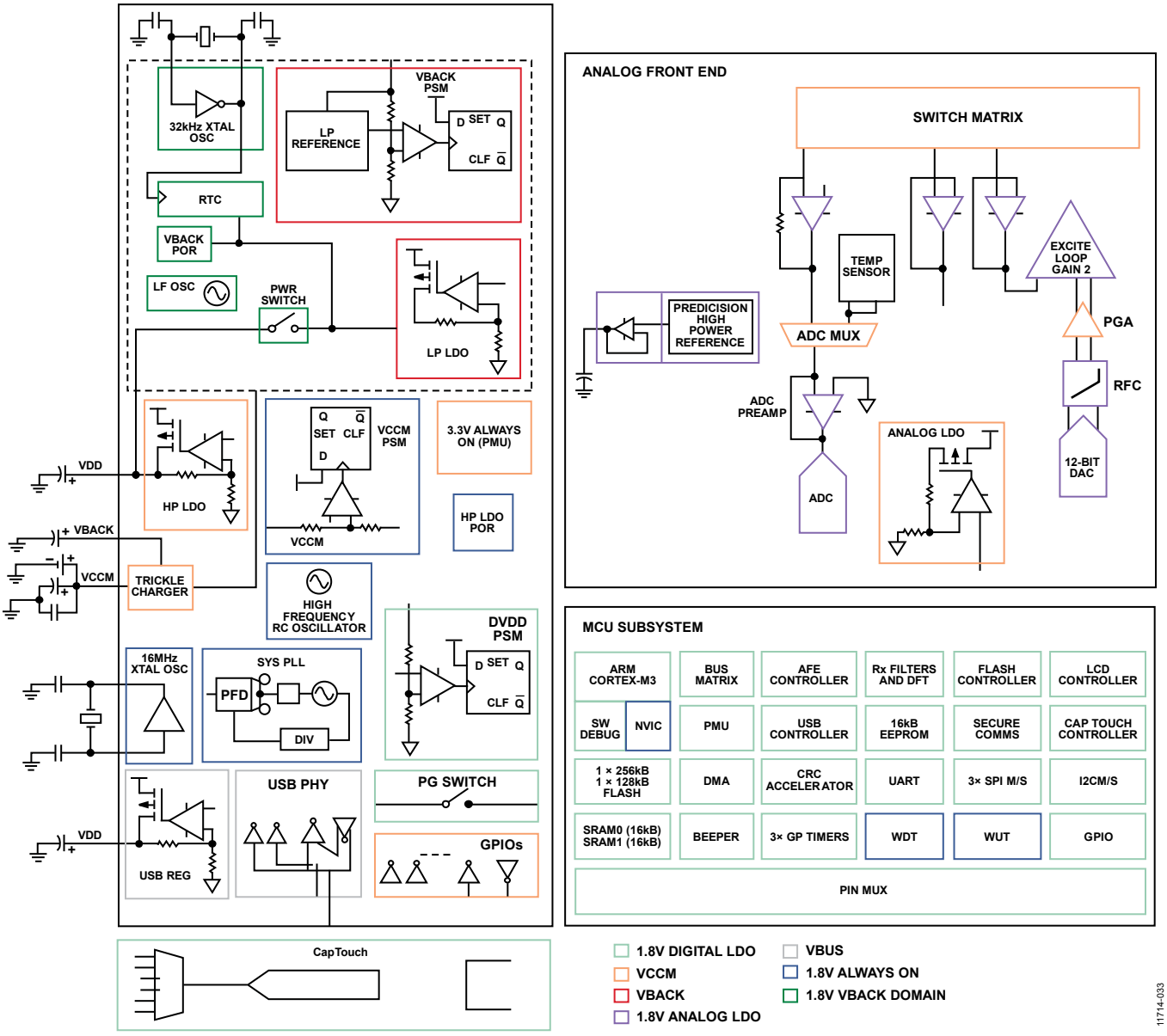


Figure 34. ADuCM350 Block Diagram with Power Supply Legend

POWER SUPPLY SOURCES

The ADuCM350 is designed to be powered by an external coin battery as a primary supply source. In case of primary supply failure (that is, the battery being removed), the system provides a secondary input supply source to keep the RTC block powered.

To switch between the two supply sources, the ADuCM350 includes an input switch controlled by an internal voltage supervisor. The switch connects the primary source, VCCM, or the supply reservoir, VBACK, to the internal supply net, VBACK_INT.

The secondary supply source is provided by a supercapacitor, and the VCCM source supplies the trickle charger for the supercapacitor. Table 108 shows the input range for each supply source.

Table 108. Power Supply Specifications

Pin Name	Description	Min	Max	Unit
VCCM_ANA and VCCM_DIG Pins (Common VCCM Domain)	ADuCM350 standby operation	2	3.6	V
	ADuCM350 AFE measurement	2.5	3.6	V
VBACK	Backup mode only	1.62	3.6	V

POWER SUPPLY CONSIDERATIONS

There are five independent supplies in the system that can be present. In all of the following cases, the presence of a supply, independent of any other supply, does not cause damage or excessive current draw:

- The meter is powered from a CR2032 battery.
- The USB regulator only powers the PHY block.
- The meter can be plugged into the USB when the CR2032 can be fully or partially depleted.
- No measurement function or USB communication can occur unless the VCCM_x pins have sufficient potential.

TRICKLE CHARGER

A smart diode in the trickle charger block ensures that whenever VCCM is higher than the supercapacitor, the battery compensates for this difference by providing the supercapacitor with additional power. This smart diode is an active circuit that ensures a forward drop of only tens of millivolts, but at the same time, limits the supercapacitor charge current to the ~500 µA seen in many supercapacitor data sheet recommendations. The active circuitry requires only ~100 nA itself. A simplified diagram is shown in Figure 35.

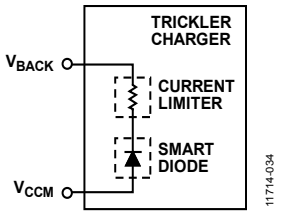


Figure 35. Trickle Charger

POWER SUPPLY MODES

The ADuCM350 provides three internal linear regulators (LDO):

- High power LDO (HP_LDO)
- Low power LDO (LP_LDO)
- Analog LDO (ALDO)

These regulators provide support for up to five power modes, divided into the following three categories, depending on the consumption of the ADuCM350 (see Power Management Unit section for more details).

- High/medium consumption: the system is powered by the battery, and the high power LDO is used to supply power. This category includes active, core sleep, and system sleep modes.
- Low consumption: the system is powered by the battery. The low power LDO is connected to the DVDD supply it supplies the circuits that are active, keeps any selected memory contents, and retains the state of digital sequential logic. This category consists of the ADuCM350 in hibernate mode.
- Backup consumption: the system is powered by the supercapacitor. The low power LDO is enabled, but only the RTC clock is active. DVDD supply is automatically disconnected to reduce power. All the data that has not been stored is lost. This category consists of the ADuCM350 in backup mode.

HIGH POWER LDO

The high power LDO is only enabled if the Cortex-M3 is in high/medium power mode. It regulates the voltage directly from the battery. This LDO is the main source of power for the DVDD supply that provides supply for the digital core, memory, peripherals, supporting clock generation, and supply monitoring circuits. It can operate from 2.0 V to 3.6 V (battery). The dropout voltage is 150 mV. It provides a typical output voltage of 1.8 V and can source up to 30 mA. It needs a typical external capacitance of 0.47 µF and an ESR of ≤0.05 Ω to maintain stability; a block diagram is shown in Figure 36.

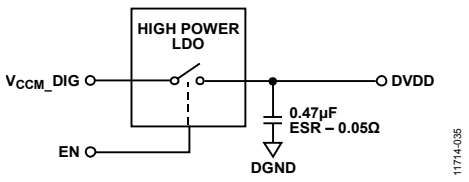


Figure 36. LDO Block Diagram

LOW POWER LDO

The low power LDO is enabled all the time. It regulates the voltage from the internal V_{BACK_INT} net. This LDO powers different circuits depending on the supply power mode, as shown in Figure 37.

- High/medium power: the LDO exclusively powers the RTC circuit block.
- Low power: DVDD supply is switched and controlled by this LDO. Most of the digital systems are internally disabled, and the low power LDO provides power to the remaining circuits.
- Backup: in this mode, only the RTC block remains powered.

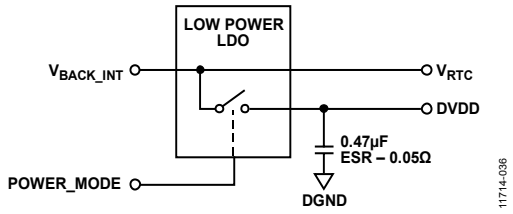


Figure 37. Low Power LDO with Power Switch

ANALOG LDO

The ALDO is used to supply the precision analog circuits, including the ADC, DAC, precision reference, temperature sensor, and instrumentation amplifier loop. This regulator is directly controlled by the Cortex-M3 ALDO bit, Bit 4 of the AFE_CFG register, and can be enabled only if the system is in active mode.

When the ALDO is powered up, a current-limit enable bit, the ALDOILIMIT_EN bit in the AFE_CFG register, is required to be set to minimize the current pulled from the battery during power-up (similar to the VREF buffer). After the analog front-end LDO is powered up, ALDOILIMIT_EN can be cleared.

See the Analog Front-End Interface section for more details on how to enable the ALDO.

The LDO requires an external capacitor of 0.47 µF with an ESR of ≤0.05 Ω typical. The dropout voltage is 150 mV. It can operate from 2.0 V to 3.6 V (battery).

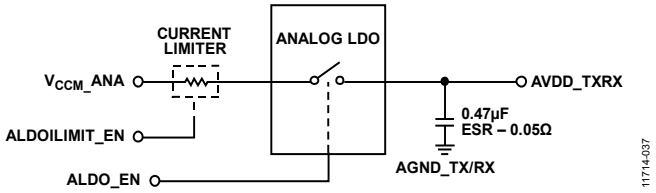


Figure 38. LDO Block Diagram

SUPPLY MONITORS

The ADuCM350 includes four power supply monitors (PSMs) to detect a supply failure.

- VBACK PSM
- VCCM PSM
- DVDD PSM
- VRTC PSM

VBACK PSM

The external battery, VCCM supply, is accurately monitored by a low power monitor, VBACK PSM. The VBACK PSM places the ADuCM350 into backup mode when VCCM supply falls below a defined threshold. The VBACK PSM is always on.

This supply monitor ensures that the VBACK switch swaps between supplies if the voltage drops below a configurable VBACKTRIP value (VBACKCON[11:8], 1.6 V to 3.1 V), as shown in Figure 39.

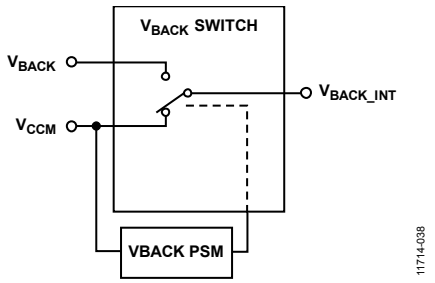


Figure 39. VBACK Switch

The VBACK PSM also monitors when the battery voltage rises with a new battery insertion, restoring the VBACK switch from the backup mode back to the active mode, as shown in Figure 40. This threshold voltage is programmable in 100 mV steps from 1.8 V to 3.3 V, using the VBACKRESTORE bits (VBACKCON[3:0]). To avoid undefined operation, the VBACKRESTORE threshold must always be set at least 200 mV greater than the VBACKTRIP threshold.

Note that a nondefault VBACKRESTORE setting has an effect only if the battery is replaced before the supercapacitor is depleted. After the supercapacitor depletes, and VBACK drops below the VBACK POR level, and all internal settings are reset to the hardware default.

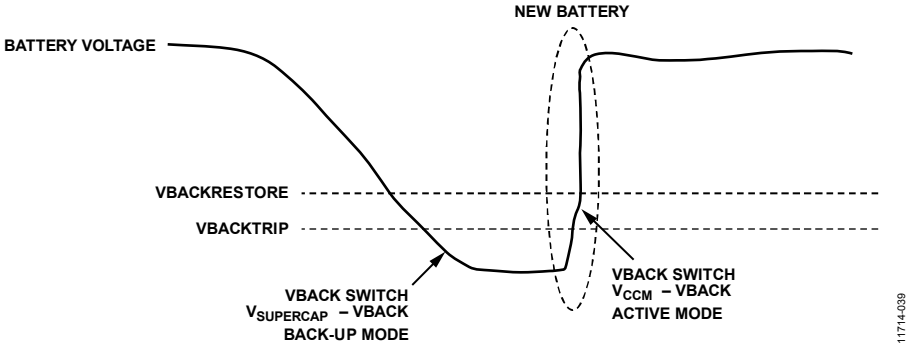


Figure 40. VBACK Switch Threshold Levels ($V_{SUPERCAP}$ is the voltage into the VBACK pin.)

When there is a complete supply source failure, the supply must be reestablished, and then the VBACK switch defaults to connecting the internal VBACK_INT node to the VCCM supply to ensure a correct power-up.

VCCM PSM

A second PSM is provided for a fast battery check, ensuring a correct voltage level before any high current demand block is enabled. The VCCM PSM can also be used to monitor the battery level to ensure a correct system suspension if the system needs to be placed in backup mode before the battery drops below a defined threshold,

This PSM must be enabled manually, due to the higher current consumption, to allow faster level detection. Additionally, by enabling the LOADENABLE bit (Bit 0) in the VCCMCON register, VCCM PSM allows an internal load of 820 Ω to be connected directly to the battery, which allows for battery strength estimation (see Figure 41).

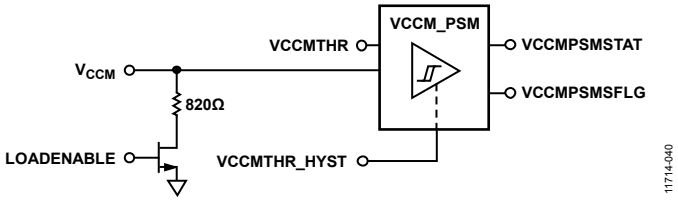


Figure 41. VCCM Power Supply Monitor

The VCCM PSM threshold is programmable in 100 mV steps from 1.7 V to 3.2 V using the VCCMTHR bits (VCCMCON[7:4]).

The respective hysteresis is adjustable using the VCCMTHR_HYST bits (VCCMCON[9:8]). Note that the actual trip point is the threshold setting plus or minus the hysteresis value.

There are two bits in PSMCON that provide the status of the battery.

- VCCMPSMSTAT provides the actual status of VCCM PSM. The VCCMPSMSTAT bit is invalid if the VCCM PSM is not enabled.
- VCCMPSMFLG is a sticky flag that detects if the battery has recovered from a previous low voltage condition. The VCCMPSMFLG is cleared only after enabling the NMI interrupt source.

The general operation is as follows:

1. A resistor divider and comparator are used to monitor the VCCM supply. The tap point on the resistor divider is programmable in 100 mV steps from 1.7 V to 3.2 V, using the VCCMTHR bits (VCCMCON[7:4]). By programming VCCMTHR and monitoring VCCMPSMSTAT, the user can search for the approximate battery voltage. Note that if the battery voltage is very close to VCCMTHR, it may take up to 200 µs for VCCMPSMSTAT to change state.
2. By connecting an 820 Ω resistor (±10%) across the supply and rechecking the battery voltage, the user can assess the strength of the battery. This allows the user to determine, for example, if the battery voltage drops 200 mV after the load is applied.
3. The resistive load can then be disconnected. The Cortex-M3 must monitor the status of the flag registers, VCCMCON[7:6], and decide if the battery level is sufficient or not. If not, the device must store all relevant information and prepare to enter backup mode.
4. If the battery is deemed strong enough to continue, the user can then set VCCMTHR to a low level, and then enable the VCCM PSM interrupt by clearing VCCMPSMIRQ_OFF (PSMCON[4]). The processor receives an interrupt if VCCM becomes too low for any reason.

VCCM PSM is powered by DVDD supply. This monitor can be enabled in hibernate mode if desired (slightly increasing the power consumption by ~1.2 µA). See the Power Management Unit section for more details.

Note that the default powered off state of this PSM is 0, which also signifies the battery being too low. If the NMI is enabled for the VCCM PSM, a delay time of 200 µs is required for the PSM to settle.

DVDD PSM

The DVDD supply includes a PSM to detect if the voltage has dropped below the limits for a correct flash memory write/read operation. This PSM is disabled in hibernate mode.

The threshold is programmed by Analog Devices. The interrupt for this PSM is disabled by default and can be enabled if desired using the DVDDPSMIRQ_OFF bit (Bit 0) in the PSMCON register. If the NMI is enabled for the DVDD PSM, a delay time of 200 µs is required for the PSM to settle. The interrupt source must be manually cleared by setting the respective interrupt bit.

The current status of the DVDD PSM can be read using the DVDDPSMSTAT bit (Bit 3) in the PSMCON register, or if the PSM has been triggered previously, can be read using the DVDDPSMFLG bit (Bit 2) in the PSMCON register.

VRTC PSM

The VRTC net powers the RTC oscillators and must be monitored continuously to ensure a correct voltage; therefore, the PSM is always on. If the VRTC supply falls below the programmable limits, a supply failure is generated and the RTC clock is reset.

The limit can be set up using the VLOTRIP bits (Bits[5:4]) in the VBACKCON register. The limit is configurable in 50 mV steps from 1.55 V to 1.70 V. A typical voltage discharge example is shown in Figure 42.

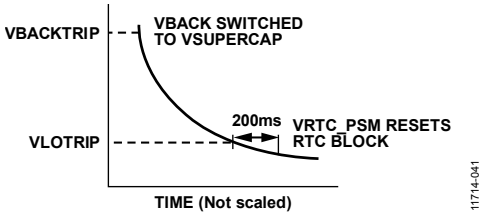


Figure 42. Typical Battery Voltage Discharge

The PSM provides programmable hysteresis using the VLTRIP_HYST bits (Bits[7:6]) in the VBACKCON register. Note that the actual trip point is the VLOTRIP setting plus or minus the hysteresis value. A VLO trip point duration of typically 200 ms (t_{VLO}) is counted out before the RTCFAIL flag bit is set.

The RTCFAIL set bit (Bit 0 of the RTCSR0 register) indicates that the time RTC clocks have been reset. Upon insertion of a new battery, the Cortex-M3 processor can read the status of the flag bit and take the appropriate action.

POWER-ON RESET CIRCUITS

The ADuCM350 provides two POR circuits to generate a correct initialization after any supply failure.

The VBACK POR circuit accurately monitors the VBACK_INT net, ensuring that the reset pulse is held low until the unregulated supply is above the minimum supply level of typically 1.8 V. When a complete supply failure has been detected, this POR initializes the ADuCM350 after the VBACK supply is restored. The POR trips at a typical value of 1.48 V on the way down, that is, typically below VLOTRIP.

The DVDD POR resets the Cortex-M3 if the DVDD voltage drops below 1.5 V when the system is active (HP_LDO enabled) or drops below typically 1.35 V when the system is in another mode. The POR releases the reset when DVDD rises above 1.55 V.

The DVDD POR circuit has two modes, depending on the PMU mode. If HP_LDO is enabled, the POR is operating in a fast mode, ensuring accurate threshold detection. If HP_LDO is disabled, the POR is in a low power mode, providing slower voltage detection.

POWER CASES

There are four case scenarios that must be considered: new battery insertion, changing from active mode to hibernate mode, changing from hibernate mode to active mode, and battery removal.

The CR2032 battery has significant output impedance, especially at low temperature. Excessive current draw at startup may cause a power supply drop, causing a power-on reset event. To prevent an incorrect power-up, a specific sequence is implemented to ensure that the current demand remains limited until the battery insertion transition has been completed.

New Battery Insertion

When a new battery is placed in a new system, the VBACK switch defaults to connecting the internal VBACK_INT node to the VCCM supply. The battery powers the VBACK POR (power-on reset) circuit, which generates a reset signal pulse for 32 ms based on the 32 kHz clock. The timer is used to guarantee that the internal reference is stable to ensure that the various PSMs are reliable before further supply decisions are made.

After the timer overflows, HP_LDO is enabled if VCCM is at or above the default VBACKRESTORE threshold (2.4 V). The internal DVDD power level good signal (see DVDD_PGOOD in Figure 43) initiates a 4 ms timer, after which the reset signal for the ADuCM350 is released, placing the system in active mode, as shown in Figure 43. Note the spike in I (VCCM) due to the regulator startup.

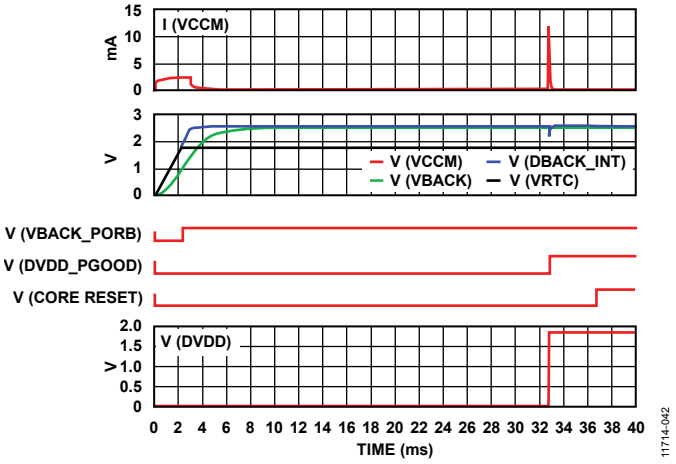


Figure 43. Power-Up Sequence with a New Battery

From Active Mode to Hibernate Mode

HP_LDO is disabled, and LP_LDO supplies energy to the RTC block and other hibernate mode blocks (see the Power Management Unit section for further details). A plot of different internal signals is shown in Figure 44. Note that the DVDD supply supplying the always-on blocks stays high while the DVDD supply supplying the majority of blocks (DVDD_VIRT) and nonretainable SRAM memory (DVDD_SRAM) goes low. By setting PWRMOD[1:0] = 11, the device goes into hibernate mode.

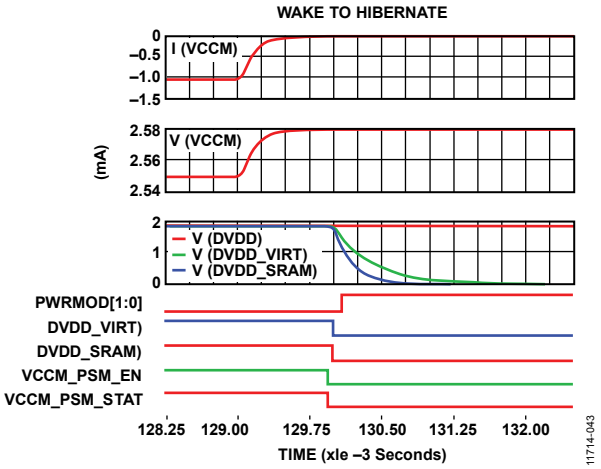


Figure 44. Active to Hibernate Mode (VCCM_PSM_EN and VCCM_PSM_STAT = Indicators)

From Hibernate Mode to Active Mode

HP_LDO is enabled, providing energy to the digital circuitry. The LP_LDO supply is connected exclusively to the RTC block. A simulation of different internal signals is shown in Figure 45. When VCCM PSM is enabled, the VCCM PSM threshold levels are checked with the battery loaded and unloaded to define the optimum settings.

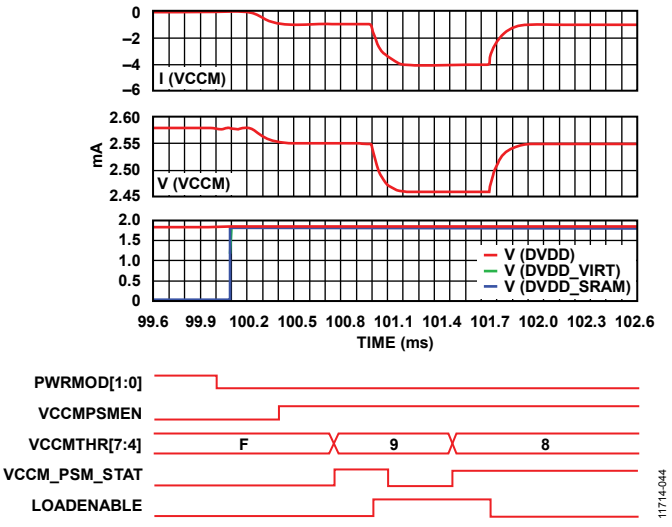


Figure 45. Hibernate to Active Mode

Battery Removal

If the battery is removed, the ADuCM350 can be in one of the four power modes controlled by the PMU.

Active, Core Sleep, and System Sleep Modes

For active, core sleep, and system sleep modes, the sequence to remove the battery is as follows:

1. VCCM PSM generates an interrupt.
2. VBACK PSM switches VBACK_INT to VBACK.
3. HP_LDO uses energy from the external decoupling capacitor connected to the battery.
4. In a reasonable amount of time (depending on the PMU mode and peripherals enabled), the external capacitor is discharged until HP_LDO can no longer keep DVDD at 1.8 V. When DVDD starts to fall, the DVDD PSM can issue another interrupt.
5. Eventually, the DVDD POR generates a hard Cortex-M3 reset.
6. Ideally, before the external decoupling capacitor is completely depleted, the user must disconnect as many peripherals as possible from the Cortex-M3 and complete the necessary system tasks before placing the device in hibernate mode.

Typical plots are shown in Figure 46 and Figure 47 for different external capacitors.

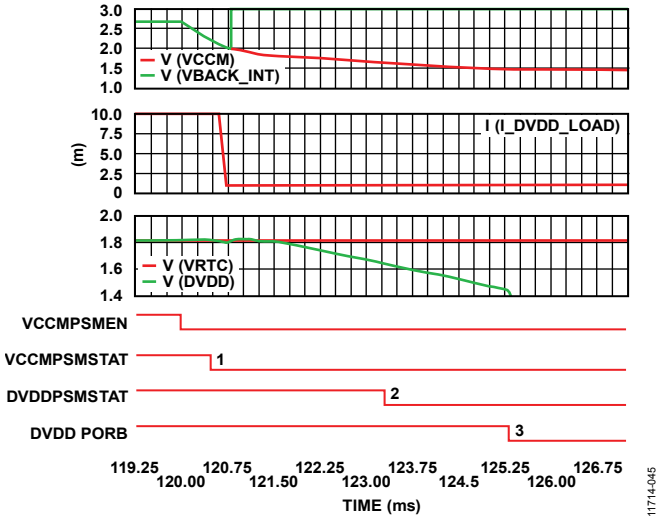


Figure 46. Removing Battery with a 10 µF Capacitor in Active Mode (DVDD PORB = Power-On Reset Block)

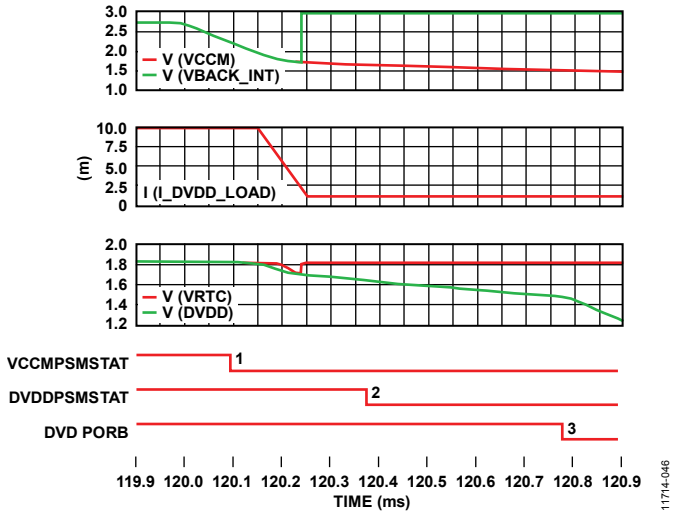


Figure 47. Removing Battery with a 2 µF Capacitor in Active Mode

Hibernate Mode

With a 10 µF VCCM decoupling capacitor and a 1.4 µA load (VCCM PSM disabled), it may take 3.6 sec to sink from 2.5 V to a VBACKTRIP of 2.0 V. Note that a very small supercapacitor is used for this example to show fast decay of VBACK_INT.

In hibernate mode, the sequence to remove the battery is as follows:

1. VBACK PSM switches VBACK_INT to VBACK.
2. LP_LDO is disconnected from DVDD supply.
3. In hibernate mode, the VBACK PSM immediately forces the Cortex-M3 into reset, without waiting for DVDD to collapse (it is assumed that there is not be enough energy left in the decoupling capacitor to allow the Cortex-M3 to wake up properly and then service the interrupts).

A typical signal plot is shown in Figure 48.

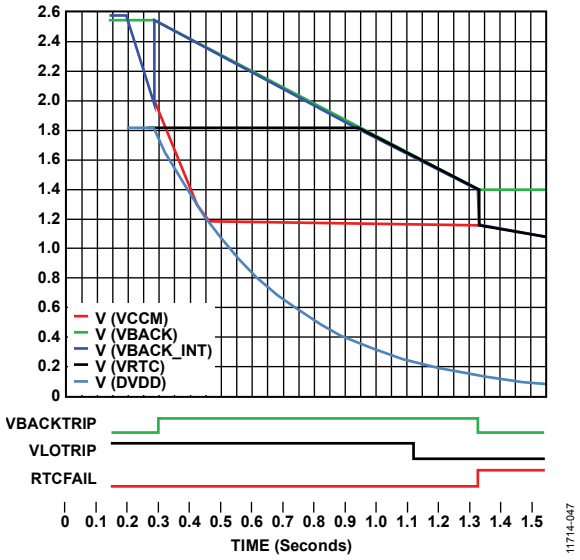


Figure 48. Removing Battery During Hibernate

Supercapacitor Collapses

When removing the battery, if the VRTC falls below the threshold programmed in VRTC PSM, the RTC block is reset after 200 ms, as shown in Figure 49.

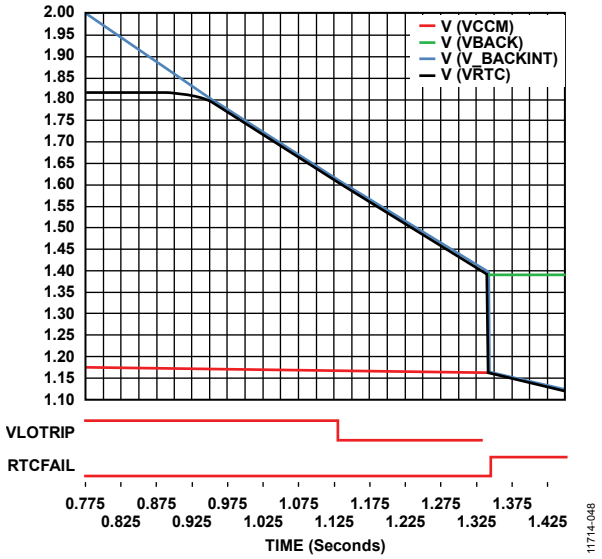


Figure 49. VRTC Generates Reset Pulse

POWER SUPPLIES MEMORY MAPPED REGISTERS

PSM Register Map

Table 109. PSM Register Summary

Address	Name	Description	Reset	RW
0x40002408	PSMCON	PSM configuration register	0x0011	RW
0x40002488	VCCMCON	VCCM control and status register	0x0080	RW
0x4000248C	VBACKCON	VBACK control and status register	0x0416	RW

PSM Configuration Register

Address: 0x40002408, Reset: 0x0011, Name: PSMCON

Table 110. Bit Descriptions for PSMCON

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	VCCMPSMSTAT	VCCM PSM current status. 0: VCCM power level is good. 1: VCCM voltage lower than the threshold.	0x0	R
6	VCCMPSMFLG	VCCM PSM sticky flag. 0: VCCM power level is good. 1: VCCM voltage lower than the threshold.	0x0	W1C
5	RESERVED	Reserved.	0x0	R
4	VCCMPSMIRQ_OFF	Disable VCCM PSM to generate an NMI interrupt. 0: VCCM PSM is enabled and generates an interrupt. 1: VCCM PSM is disabled and does not generate an interrupt.	0x1	RW
3	DVDDPSMSTAT	DVDD PSM current status. 0: DVDD power level is good. 1: DVDD voltage is lower than the threshold.	0x0	R
2	DVDDPSMFLG	DVDD PSM sticky flag. Write 1 to clear this status bit. 0: DVDD power level is good. 1: DVDD voltage is lower than the threshold.	0x0	W1C
1	RESERVED	Reserved.	0x0	R
0	DVDDPSMIRQ_OFF	Disable DVDD PSM to generate an NMI interrupt. 1: DVDD PSM is disabled and does not generate interrupt. 0: DVDD PSM is enabled and generates an interrupt.	0x1	RW

VCCM Control and Status Register

Address: 0x40002488, Reset: 0x0080, Name: VCCMCON

Table 111. Bit Descriptions for VCCMCON

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	RW
[9:8]	VCCMTHR_HYST	VCCM threshold hysteresis. 00: 10 mV. 01: 20 mV. 10: 50 mV. 11: 100mV.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[7:4]	VCCMTHR	VCCM threshold adjust. 0000: 1.7 V. 0001: 1.8 V. 0010: 1.9 V. 0011: 2.0 V. 0100: 2.1 V. 0101: 2.2 V. 0110: 2.3 V. 0111: 2.4 V. 1000: 2.5 V. 1001: 2.6 V. 1010: 2.7 V. 1011: 2.8 V. 1100: 2.9 V. 1101: 3.0 V. 1110: 3.1 V. 1111: 3.2 V.	0x8	RW
[3:2]	RESERVED	Reserved.	0x0	R
1	VCCMPSMEN	VCCM power supply monitoring enable. 0: VCCM PSM is disabled. 1: VCCM PSM is enabled.	0x0	RW
0	LOADENABLE	Enabling 820 Ω load.	0x0	RW

VBACK Control and Status Register

Address: 0x4000248C, Reset: 0x0416, Name: VBACKCON

Table 112. Bit Descriptions for VBACKCON

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
[11:8]	VBACKTRIP	VBACK trip point adjust. 0000: 1.6 V. 0001: 1.7 V. 0010: 1.8 V. 0011: 1.9 V. 0100: 2.0 V. 0101: 2.1 V. 0110: 2.2 V. 0111: 2.3 V. 1000: 2.4 V. 1001: 2.5 V. 1010: 2.6 V. 1011: 2.7 V. 1100: 2.8 V. 1101: 2.9 V. 1110: 3.0 V. 1111: 3.1 V.	0x4	RW
[7:6]	VLTRIP_HYST	VLO trip hysteresis. 00: 25 mV. 01: 50 mV. 10: 75 mV. 11: 100 mV.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[5:4]	VLOTRIP	VLO trip level. 00: 1.55 V. 01: 1.60 V. 10: 1.65 V. 11: 1.70 V.	0x1	RW
[3:0]	VBACKRESTORE	VBACK restore level. 0000: 1.8 V. 0001: 1.9 V. 0010: 2.0 V. 0011: 2.1 V. 0100: 2.2 V. 0101: 2.3 V. 0110: 2.4 V. 0111: 2.5 V. 1000: 2.6 V. 1001: 2.7 V. 1010: 2.8 V. 1011: 2.9 V. 1100: 3.0 V. 1101: 3.1 V. 1110: 3.2 V. 1111: 3.3 V.	0x6	RW

SYSTEM INTERRUPTS AND EXCEPTIONS

ARM CORTEX-M3 EXCEPTIONS

The Cortex-M3 supports a number of system exceptions and peripheral interrupts. Exception 1 to Exception 15 are system exceptions and are described in Table 113.

Table 113. List of System Exceptions

Number	Type	Priority	Description
1	Reset	−3 (highest)	Any reset
2	NMI	−2	Nonmaskable interrupt connected to power supply monitor
3	Hard fault	−1	All fault conditions, if the corresponding fault handler is not enabled
4	Memory management fault	Programmable	Memory management fault; access to illegal locations
5	Bus fault	Programmable	Prefetch fault, memory access fault, and other address/memory related faults
6	Usage fault	Programmable	Exceptions such as undefined instruction executed or illegal state transition attempt
7 to 10	Reserved	Not applicable	
11	SVCALL	Programmable	System service call with SVC instruction
12	Debug monitor	Programmable	Debug monitor (breakpoint, watch point, or external debug requests)
13	Reserved	Not applicable	
14	PENDSV	Programmable	Pendable request for system service
15	SYSTICK	Programmable	System tick timer

NESTED VECTORED INTERRUPT CONTROLLER (NVIC)

Interrupts are controlled by the NVIC, and eight levels of priority are available. Only a limited number of interrupts can wake up the device from hibernate mode. These are described in the Power Management Unit section in detail.

When the device is woken up from CORE_SLEEP, SYS_SLEEP, or hibernate mode, it always returns to full active mode.

Table 114. Interrupt Sources

		Wake-Up From		
Position No.	Vector	CORE_SLEEP	SYS_SLEEP	Hibernate
0	Wake-up timer	Yes	Yes	Yes
1	External Interrupt 0 (P4.0)	Yes	Yes	Yes
2	External Interrupt 1 (P0.0)	Yes	Yes	Yes
3	External Interrupt 2 (P0.1)	Yes	Yes	Yes
4	External Interrupt 3 (P0.2)	Yes	Yes	Yes
5	External Interrupt 4 (P0.3)	Yes	Yes	Yes
6	External Interrupt 5 (P0.4)	Yes	Yes	Yes
7	External Interrupt 6 (P0.5)	Yes	Yes	Yes
8	External Interrupt 7 (P3.4)	Yes	Yes	Yes
9	External Interrupt 8 (P0.10)	Yes	Yes	Yes
10	Watchdog timer	Yes	Yes	Yes
11	General-Purpose Timer 0	Yes	No	No
12	General-Purpose Timer 1	Yes	No	No
13	Flash Controller 0	Yes	No	No
14	UART	Yes[1]	No	No
15	SPI0	Yes[1]	No	No
16	SPI-H	Yes[1]	No	No
17	I2C0 slave	Yes[1]	No	No
18	I2C0 master	Yes[1]	No	No
19	DMA error	Yes	No	No
20	DMA Channel 0 done	Yes	No	No
21	DMA Channel 1 done	Yes	No	No
22	DMA Channel 2 done	Yes	No	No

Position No.	Vector	Wake-Up From		
		CORE_SLEEP	SYS_SLEEP	Hibernate
23	DMA Channel 3 done	Yes	No	No
24	DMA Channel 4 done	Yes	No	No
25	DMA Channel 5 done	Yes	No	No
26	DMA Channel 6 done	Yes	No	No
27	DMA Channel 7 done	Yes	No	No
28	DMA Channel 8 done	Yes	No	No
29	DMA Channel 9 done	Yes	No	No
30	DMA Channel 10 done	Yes	No	No
31	DMA Channel 11 done	Yes	No	No
32	DMA Channel 12 done	Yes	No	No
33	DMA Channel 13 done	Yes	No	No
34	DMA Channel 14 done	Yes	No	No
35	DMA Channel 15 done	Yes	No	No
36	USB wake up	Yes	Yes	Yes
37	USB	Yes	No	No
38	USB DMA	Yes	No	No
39	I^2S	Yes[1]	No	No
40	General-Purpose Timer 2	Yes	No	No
41	Flash Controller 1	Yes	No	No
42	SPI1	Yes[1]	No	No
43	Real-time clock	Yes	Yes	Yes
44	Reserved	Not applicable	Not applicable	Not applicable
45	Beeper	Yes	No	No
46	LCD controller	Yes	No	No
47	GPIO IntA	Yes	No	No
48	GPIO IntB	Yes	No	No
49	Reserved	Not applicable	Not applicable	Not applicable
50	AFE controller (analog capture interrupt)	Yes	Yes	No
51	AFE controller (analog generation interrupt)	Yes	Yes	No
52	AFE controller (command FIFO interrupt)	Yes	Yes	No
53	AFE controller (data FIFO interrupt)	Yes	Yes	No
54	CapTouch	Yes	Yes	No
55	General-purpose flash controller (EEPROM)	Yes	No	No
56	Crystal oscillator	Yes	Yes	No
57	PLL	Yes	Yes	No
58	Random bit generator	Yes	No	No
59	Parallel display interface	Yes	No	No
60	Flash parity error interrupt	Yes	No	No

[1] Corresponding UCLK is required to generate the interrupt.

Internally, the highest user-programmable priority (0) is treated as the fourth priority—after a reset, NMI, and a hard fault. Note that 0 is the default priority for all programmable priorities.

If the same priority level is assigned to two or more interrupts, their hardware priority (the lower the position number) determines the order in which the processor activates them. For example, if both SPI0 and SPI1 are Priority Level 1, SPI0 has higher priority.

See the Exceptions section and the Nested Vector Interrupt Controller section in the ARM Cortex-M3 Technical Reference Manual for more information on the exceptions and interrupts.

HANDLING INTERRUPT REGISTERS

In the interrupt service routine (ISR) for any interrupt source, the first action usually taken is clearing the interrupt source. In case of write 1 to clear interrupts, the interrupt status register must be written to clear the interrupt source. Due to internal delays in the bus matrix, this write may be delayed before it reaches the destination.

Meanwhile, the ISR execution might have been completed, and there is a possibility of mistaking the previous interrupt for a second one. Therefore, it is always recommended to ensure that the interrupt source has been cleared before exiting the ISR, which can be done by reading the interrupt status again at the end of the ISR.

EXTERNAL INTERRUPT CONFIGURATION

Nine external interrupts are implemented. These external interrupts can be separately configured to detect any combination of the following types of events:

- Rising edge: the logic detects a transition from low to high and generates a pulse. Only one pulse is sent to the Cortex-M3 per rising edge.
- Falling edge: the logic detects a transition from high to low and generates a pulse. Only one pulse is sent to the Cortex-M3 per falling edge.
- Rising or falling edge: the logic detects a transition from low to high or high to low and generates a pulse. Only one pulse is sent to the Cortex-M3 per edge.
- High level: the logic detects a high level. The appropriate interrupt is asserted and sent to the Cortex-M3. The interrupt line is held asserted until the external source deasserts. The high level must be maintained for one core clock cycle minimum to be detected.
- Low level: the logic detects a low level. The appropriate interrupt is asserted and sent to the Cortex-M3. The interrupt line is held asserted until the external source deasserts. The low level must be maintained a minimum of one core clock cycle to be detected.

The external interrupt detection unit block is in the always on section and allows the external interrupt to wake up the device when in hibernate mode.

Similarly, the USB can wake up the system from hibernate mode. Wake-up detection logic is added to the USB controller to check the status of the USB bus, including DP, DM, and VBUS. Wake-up conditions on the bus generate a USB wake-up interrupt that wakes up the device much in the same way as external interrupts.

SYSTEM INTERRUPTS MEMORY MAPPED REGISTERS

The interrupt unit consists of five MMRs contained in the always on block and based at Address 0x40002400.

System Interrupts Register Map

Table 115. Interrupt Detection Unit Register Summary

Address	Name	Description	Reset	RW
0x40002420	EI0CFG	External Interrupt Configuration Register 0.	0x0000	RW
0x40002424	EI1CFG	External Interrupt Configuration Register 1.	0x0000	RW
0x40002428	EI2CFG	External Interrupt Configuration Register 2.	0x0000	RW
0x4000242C	RESERVED	Reserved.		
0x40002430	EICLR	External interrupt clear register.	0x0000	RW
0x40002434	NMICLR	Nonmaskable interrupt clear register.	0x0000	RW
0x40002438	USBWKSTAT	USB wake-up status register.	0x0000	R
0x40002440	RSTSTA	Reset status/clear register.	0x0000	RW

External Interrupt Configuration 0 Register

Address: 0x40002420, Reset: 0x0000, Name: EI0CFG

Table 116. Bit Descriptions for EI0CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ3EN	External Interrupt 3 enable bit. 0: External Interrupt 3 disabled. 1: External Interrupt 3 enabled.	0x0	RW
[14:12]	IRQ3MDE	External Interrupt 3 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

Bits	Bit Name	Description	Reset	Access
11	IRQ2EN	External Interrupt 2 enable bit. 0: External Interrupt 2 disabled. 1: External Interrupt 2 enabled.	0x0	RW
[10:8]	IRQ2MDE	External Interrupt 2 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
7	IRQ1EN	External Interrupt 1 enable bit. 0: External Interrupt 0 disabled. 1: External Interrupt 0 enabled.	0x0	RW
[6:4]	IRQ1MDE	External Interrupt 1 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
3	IRQOEN	External Interrupt 0 enable bit. 0: External Interrupt 0 disabled. 1: External Interrupt 0 enabled.	0x0	RW
[2:0]	IRQ0MDE	External Interrupt 0 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

External Interrupt Configuration 1 Register

Address: 0x40002424, Reset: 0x0000, Name: EI1CFG

Table 117. Bit Descriptions for EI1CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ7EN	External Interrupt 7 enable bit. 0: External Interrupt 7 disabled. 1: External Interrupt 7 enabled.	0x0	RW
[14:12]	IRQ7MDE	External Interrupt 7 Mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

Bits	Bit Name	Description	Reset	Access
11	IRQ6EN	External Interrupt 6 Enable bit. 0: External Interrupt 6 disabled. 1: External Interrupt 6 enabled.	0x0	RW
[10:8]	IRQ6MDE	External Interrupt 6 Mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
7	IRQ5EN	External Interrupt 5 Enable bit. 0: External Interrupt 5 disabled. 1: External Interrupt 5 enabled.	0x0	RW
[6:4]	IRQ5MDE	External Interrupt 5 Mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
3	IRQ4EN	External Interrupt 4 Enable bit. 0: External Interrupt 4 disabled. 1: External Interrupt 4 enabled.	0x0	RW
[2:0]	IRQ4MDE	External Interrupt 4 Mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

External Interrupt Configuration 2 Register

Address: 0x40002428, Reset: 0x0000, Name: EI2CFG

Table 118. Bit Descriptions for EI2CFG

Bits	Bit Name	Description	Reset	Access
15	USBVBUSEN	USB VBUS detection enable bit. 0: USB VBUS detection disabled. 1: USB VBUS detection enabled.	0x0	RW
[14:12]	USBVBUSMDE	USB VBUS detection mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

Bits	Bit Name	Description	Reset	Access
11	USBDMEN	USB DM detection enable bit. 0: USB DM detection disabled. 1: USB DM detection enabled.	0x0	RW
[10:8]	USBDMMDE	USB DM detection mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
7	USBDPEN	USB DP Detection Enable bit. 0: USB DP detection disabled. 1: USB DP detection enabled.	0x0	RW
[6:4]	USBDPMDE	USB DP detection mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
3	IRQ8EN	External Interrupt 8 enable bit. 0: External Interrupt 8 disabled. 1: External Interrupt 8 enabled.	0x0	RW
[2:0]	IRQ8MDE	External Interrupt 8 Mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

External Interrupt Clear Register

Address: 0x40002430, Reset: 0x0000, Name: EICLR

Table 119. Bit Descriptions for EICLR

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	RW
11	USBVBUS	USB VBUS detection. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
10	USBDM	USB DM detection. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
9	USBDP	USB DP detection. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
8	IRQ8	External Interrupt 8. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
7	IRQ7	External Interrupt 7. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
6	IRQ6	External Interrupt 6. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
5	IRQ5	External Interrupt 5. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
4	IRQ4	External Interrupt 4. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
3	IRQ3	External Interrupt 3. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
2	IRQ2	External Interrupt 2. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
1	IRQ1	External Interrupt 1. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW
0	IRQ0	External Interrupt 0. Set to 1 to clear an internal interrupt flag. Cleared automatically by hardware.	0x0	RW

Nonmaskable Interrupt Clear Register

Address: 0x40002434, Reset: 0x0000, Name: NMICLR

Table 120. Bit Descriptions for NMICLR

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0	RW
0	CLEAR	NMI clear. Set to 1 to clear an internal interrupt flag when the NMI interrupt is set. Cleared automatically by hardware.	0x0	RW

USB Wakeup Status Register

Address: 0x40002438, Reset: 0x0000, Name: USBWKSTAT

Table 121. Bit Descriptions for USBWKSTAT

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
2	USBVBUS	USB VBUS event status. Set automatically when a USB VBUS event (configured in EI2CFG) occurs. Cleared by writing one to the USBVBUS bit in the EICLR register.	0x0	R
1	USBDM	USB DM event status. Set automatically when a USB DM event (configured in EI2CFG) occurs. Cleared by writing one to the USBDM bit in the EICLR register.	0x0	RW
0	USBDP	USB DP event status. Set automatically when a USB DP event (configured in EI2CFG) occurs. Cleared by writing one to the USBDP bit in the EICLR register.	0x0	RW

Reset Status Register

Address: 0x40002440, Reset: 0x0000, Name: RSTSTA

Table 122. Bit Descriptions for RSTSTA

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	SWRST	Software reset. Set automatically to 1 when the Cortex-M3 system reset is generated. Cleared by writing 1 to the bit.	0x0	W1C
2	WDRST	Watchdog timeout. Set automatically to 1 when a watchdog timeout occurs. Cleared by writing 1 to the bit.	0x0	W1C
1	EXTRST	External reset. Set automatically to 1 when an external reset occurs. Cleared by writing 1 to the bit.	0x0	W1C
0	POR	Power-on reset. Set automatically when a power-on reset occurs. Cleared by writing one to the bit.	0x0	W1C

RESET

RESET OPERATION

There are four kinds of resets: external, power-on, watchdog timeout, and software system reset. The software system reset is provided as part of the Cortex-M3 core.

To generate a system reset through software, the application interrupt/reset control register must be written to Address 0x05FA0004. This register is part of the NVIC register and is located at Address 0xE000ED0C. See the ARM Cortex-M3 Technical Reference Manual for more details on the software reset.

A hardware reset is performed by toggling the RESETX pin, which is active low.

The RSTSTA register indicates the source of the last reset, and RSTCLR allows clearing of the RSTSTA register. These registers can be used during a reset exception service routine to identify the source of the reset.

Table 123. Reset Types

Reset	Reset External Pins to Default State	Execute Kernel	Reset All MMRs Except RSTSTA	Reset All Peripherals	Valid SRAM	RSTSTA After Reset Event
SW	Yes[1]	Yes	Yes	Yes	Yes/no[2]	RSTSTA[3] = 1
WD	Yes	Yes	Yes	Yes	Yes/no[2]	RSTSTA[2] = 1
External Reset Pin	Yes	Yes	Yes	Yes	Yes/no[2]	RSTSTA[1] = 1
POR	Yes	Yes	Yes	Yes	No	RSTSTA[0] = 1

[1] GPIOx returns to its default state, that is, POR output. It is only low in the case of a POR event; in all other cases, it remains high.
[2] RAM is not valid in the case of a reset following a UART download.

RESET MEMORY MAPPED REGISTERS

Register Map

Table 124. Reset Register Summary

Address	Name	Description	Reset	RW
0x40002440	RSTSTA	Status register	0x00	R

Reset Status Register

Address: 0x40002440, Reset: 0x00, Name: RSTSTA

Table 125. Bit Descriptions for RSTSTA

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	SWRST	Software reset. Set automatically to 1 when the Cortex-M3 system reset is generated. Cleared by writing 1 to the bit.	0x0	W1C
2	WDRST	Watchdog timeout. Set automatically to 1 when a watchdog timeout occurs. Cleared by writing 1 to the bit.	0x0	W1C
1	EXTRST	External reset. Set automatically to 1 when an external reset occurs. Cleared by writing 1 to the bit.	0x0	W1C
0	POR	Power-on reset. Set automatically when a power-on reset occurs. Cleared by writing 1 to the bit.	0x0	W1C

DMA CONTROLLER

FEATURES

The direct memory access (DMA) controller is used to perform data transfer tasks and offload these from the processor. DMA is used to provide high speed data transfer between peripherals and memory. Data can be quickly moved by the DMA without any CPU actions, which keeps CPU resources free for other operations.

The DMA controller provides the following features:

- 16 independent DMA channels.
- Two programmable priority levels for each DMA channel.
- Each priority level arbitrates using a fixed priority that is determined by the DMA channel number.
- Each DMA channel can access a primary and/or alternate channel control data structure.
- Supports multiple transfer types:
  - Memory to memory.
  - Memory to peripheral.
  - Peripheral to memory.
- Supports multiple DMA cycle types.
  - Basic.
  - Automatic request.
  - Ping-pong.
  - Scatter gather.
- Supports multiple DMA transfer data widths (8-, 16- and 32-bit).
- Each DMA channel can have independent source and destination increment/decrement controls.

The DMA has 16 channels in total, each dedicated to managing memory access requests from peripherals. Channels are assigned as shown in Table 126.

Table 126. DMA Channel Assignment

Channel	Peripheral
0	SPI-H 0 Tx
1	SPI-H 0 Rx
2	SPI0 Tx
3	SPI0 Rx
4	SPI1 Tx
5	SPI1 Rx
6	UART Tx
7	UART Rx
8	I2C0 slave Tx
9	I2C0 slave Rx
10	I2C0 master
11	AFE Tx
12	AFE Rx
13	CRC (default) or general-purpose flash (see the GPFDMACTL Register section)
14	Parallel display interface
15	I^2S

DMA Channel 13 is shared between the CRC module and the general-purpose flash. Use of the DMA channel is exclusive to only the selected peripheral. Selection is made using the GPFDMACTL register.

GPFDMACTL REGISTER

Address: 0x40028100, Reset: 0x0000, Name: GPFDMACTL

Table 127. Bit Descriptions for GPFDMACTL

Bits	Bit Name	Description	Reset	Access
0	GPFLASH	DMA Channel 13 control. This bit controls which module is recognized by DMA Channel 13. If a module is not properly selected, the DMA controller ignores any request from that module. 0: CRC maintains control of DMA Channel 13. 1: general-purpose flash controls DMA Channel 13.	0x0	RW

Each DMA channel is connected to a dedicated hardware DMA request, and the software trigger is also supported on each channel. This configuration is done by software.

Each DMA channel has a programmable priority level default or high. Each priority level arbitrates using a fixed priority that is determined by the DMA channel number.

The DMA controller supports multiple DMA transfer data widths. However, the source and destination transfer sizes must be the same. In addition, always align the source and destination addresses to the data transfer size.

The DMA controller has a single output to indicate when an error condition (a DMA error interrupt) occurs. An error condition can occur if there is a bus error while performing a DMA transfer or if the DMA controller reads an invalid cycle control.

The DMA controller supports memory to memory, peripheral to memory, and memory to peripheral transfers and has access to flash, SRAM0, or SRAM1 as source and destination.

OPERATION

The DMA controller has two buses. One bus is connected to the system bus shared with the Cortex-M3 core, and the other bus is connected to 16-bit peripherals. The DMA request may stop the CPU access to the system bus for some bus cycles, such as when the CPU and DMA target the same destination (memory or peripheral).

The DMA controller fetches channel control data structures located in the system memory to perform data transfers. The DMA capable peripherals, when enabled to use DMA, can request the DMA controller for a transfer. At the end of the programmed number of DMA transfers for a channel, the DMA controller generates a single cycle DMA_DONE interrupt corresponding to that channel. This interrupt indicates the completion of the DMA transfer. Separate interrupt enable bits are available in the NVIC for each of the DMA channels.

Channel Control Data Structure

Every channel has two control data structures associated with it: a primary data structure and an alternate data structure. For simple transfer modes, the DMA controller uses either the primary or alternate data structure by programming the DMAALTCLR register. For more complex data transfer modes, such as ping-pong or scatter gather, the DMA controller uses both the primary and alternate data structures. Each control data structure (primary or alternate) occupies four 32-bit locations in the memory, as shown in Table 128.

Figure 50 shows the entire channel control data structure for the eight DMA channel case. It uses 512 bytes of system memory.

Table 128. Channel Control Data Structure

Offset	Name	Description
0x00	SRC_END_PTR	Source end pointer
0x04	DST_END_PTR	Destination end pointer
0x08	CHNL_CFG	Control data configuration
0x0C	RESERVED	Reserved

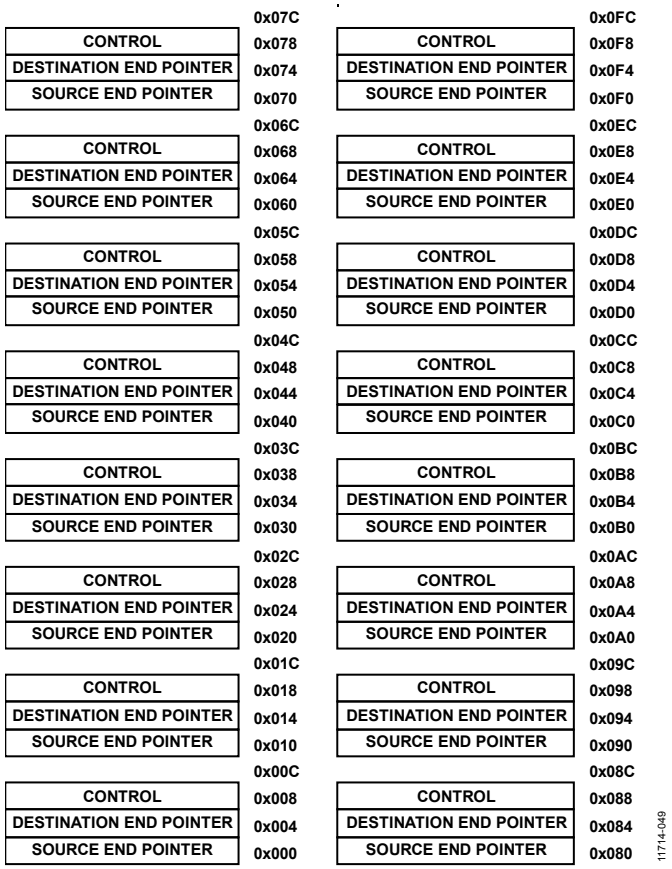


Figure 50. Memory Map for Primary DMA Structure

Before the controller can perform a DMA transfer, the data structure related to the DMA channel must be written to the designated location in system memory. The source end pointer memory location contains the end address of the source data. The destination end pointer memory location contains the end address of the destination data. The control memory location contains the channel configuration control data, which determines the source and destination data size, the number of transfers, and the number of data transfers for each arbitration.

When the DMA controller receives a request for a channel, it reads the corresponding data structure from the system memory into its internal cache. Any update to the descriptor in the system memory before the DMA controller generates the DMA_DONE interrupt may result in unexpected behavior; therefore, it is strongly recommended that the user not update the descriptor before receiving a DMA_DONE interrupt.

Source Data End Pointer

The SRC_END_PTR memory location shown in Table 129 stores the address of the last location from where data is read as part of DMA transfer. This memory location must be programmed with the end address of the source data before the controller can perform a DMA transfer. The controller reads this memory location when it starts the first DMA data transfer. Note that the DMA controller does not write to this memory location.

Table 129. Source Data End Pointer

Bit	Name	Description
[31:0]	SRC_END_PTR	The end address of the source data

Destination Data End Pointer

The DST_END_PTR memory location shown in Table 130 stores the address of the last location where data is written to as part of DMA transfer. This memory location must be programmed with the end address of the destination data before the controller can perform a DMA transfer. The controller reads this memory location when it starts the first DMA data transfer. Note that the DMA controller does not write to this memory location.

Table 130. Destination Data End Pointer

Bit	Name	Description
[31:0]	DST_END_PTR	The end address of the destination data

Control Data Configuration

For each DMA transfer, the control data configuration (CHNL_CFG) memory location provides the control information for the DMA transfer to the controller.

Table 131. Destination Data End Pointer

Bit	Name	Description
[31:30]	DST_INC	Destination address increment. The address increment depends on the source data width as follows: SRC_SIZE: byte. 00: byte. 01: halfword. 10: word. 11: no increment. Address remains set to the value that the DST_END_PTR memory location contains. SRC_SIZE: halfword. 00: reserved. 01: halfword. 10: word. 11: no increment. Address remains set to the value that the DST_END_PTR memory location contains. SRC_SIZE: word. 00: reserved. 01: reserved. 10: word. 11: no increment. Address remains set to the value that the DST_END_PTR memory location contains.
[29:28]	RESERVED	Undefined. Write as 0.
[27:26]	SRC_INC	Source address increment. The address increment depends on the source data width as follows: SRC_SIZE: byte. 00: byte. 01: halfword. 10: word. 11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains. SRC_SIZE: halfword. 00: reserved. 01: halfword. 10: word. 11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains. SRC_SIZE: word 00: reserved. 01: reserved. 10: word. 11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains.
[25:24]	SRC_SIZE	Size of the source data. 00: byte. 01: halfword. 10: word. 11: reserved.

Bit	Name	Description
[23:18]	RESERVED	Undefined. Write as 0.
[17:14]	R_POWER	R_POWER arbitrates after x DMA transfers. Note that software DMA transfers can use any value from 0000 to 1111. DMA transfers that involve peripherals must always use 0000, with the exception of the I²S drive; see the I2C Serial Interface section for further details. The operation of the DMA is indeterminate if a value other than 0000 is programmed for the DMA transfers involving peripherals. 0000: x = 1. 0001: x = 2. 0010: x = 4. 0011: x = 8. 0100: x = 16. 0101: x = 32. 0110: x = 64. 0111: x = 128. 1000: x = 256. 1001: x = 512. 1010 to 1111: x = 1024.
[13:4]	N_MINUS_1	Total number of transfers in the current DMA cycle minus 1. The 10-bit value indicates the number of DMA transfers minus one. Note that this value indicates the total number of transfers in the DMA cycle and not the total number of bytes. The possible values are as follows: 000: 1 DMA transfer. 001: 2 DMA transfers. 010: 3 DMA transfers. … 1111111111(0x3FF): 1024 DMA transfers.
3	RESERVED	Undefined. Write as 0.
[2:0]	CYCLE_CTRL	The operating mode of the DMA cycle. 000: stop (invalid). 001: basic. 010: automatic request. 011: ping-pong. 100: memory scatter gather primary. 101: memory scatter gather alternate. 110: peripheral scatter gather primary. 111: peripheral scatter gather alternate.

If any error occurs during a DMA data transfer, CHNL_CFG is written back to the system memory, and the N_MINUS_1 bit field is updated to reflect the number of transfers yet to be completed. When a full DMA cycle is complete, the CYCLE_CTRL bits become invalid to indicate the completion of the transfer.

DMA Priority

The priority of a channel is determined by its number and priority level. Each channel can have two priority levels: default or high. All channels at the high priority level have higher priority than the channels at the default priority level. At the same priority level, a channel with a lower channel number has a higher priority than a channel with a higher channel number. The DMA channel priority levels can be changed by writing to the appropriate bit in the DMAPRISET register.

DMA Transfer Timing

Each DMA transfer may vary in the amount of time it takes to execute. In the best case (that is, when all resources are available, there are no contentions, and descriptors are loaded), a DMA transfer needs five cycles to complete.

However, several factors, including the following, can result in a delay:

- The core is accessing memory.
- SPI is finishing a transfer.
- The descriptors for the AFE are bumped from the descriptor cache.

Example Use Case

In this example, the FIFO requires filling and is assumed empty. The time needed to complete a DMA transfer is as follows:

- Upon the first DMA access, there may be a delay while the previous access (for example, SPI) completes. This requires a minimum of 8 cycles.
- The AFE descriptors were bumped; therefore, the first access requires additional time to load descriptors. This requires a minimum of 8 cycles.
- The AFE receives the next six accesses (it is set as high priority), but the core is competing for access to SRAM, requiring at least an additional cycle per access (7 × 6 cycles = 42 cycles).

This results in a minimum of 58 cycles for the DMA transfer to complete; a minimum of 100 cycles is a conservative figure to use.

DMA Transfer Types

Based on how the DMA controller uses the primary and alternate control data structure and the number of requests required to complete a DMA transfer, five types of DMA transfers are supported by the DMA controller. The various types are selected by programming the appropriate values into the CYCLE_CTRL bits in the CHNL_CFG location of the control data structure. Based on CYCLE_CTRL, the following are the different DMA transfer types supported:

- Invalid
- Basic
- Auto request
- Ping-pong
- Memory scatter-gather
- Peripheral scatter-gather

Table 132 describes the various DMA transfer types that must be used by the peripherals and software DMA. Use of software DMA transfer types for peripherals and vice versa is strongly discouraged.

Table 132. Transfer Type Usage

Peripherals	Software
Basic	Auto request
Ping-Pong	Ping-pong[1]
Peripheral Scatter Gather	Memory scatter gather

[1] Software ping-pong DMA transfer operation is different from peripheral ping-pong DMA transfer operation (see the details in the Ping-Pong section).

Invalid

Invalid means that no DMA transfer is enabled for the channel. After the controller completes a DMA cycle, it sets the cycle type to invalid to prevent it from repeating the same DMA cycle. Reading an invalid descriptor for any channel generates a DMA error interrupt, and the corresponding bit in DMAINVALIDDESCCLR is set.

Basic

In this mode, the controller can be configured to use either primary or alternate data structure by programming the DMAALTCLR register for the corresponding channel. This mode is used for peripheral DMA transfers. While using this mode for peripherals, R_POWER in the CHNL_CFG register must be set to 0, which means that the peripheral must present a request for every data transfer. Following the channel being enabled, when the controller receives a request, it performs the following operations:

1. The controller performs a transfer. If the number of transfers remaining is 0, the flow continues at Step 3.
2. The controller arbitrates:
   a. If a higher priority channel is requesting service, the controller services that channel; or,
   b. If the peripheral or software signals a request to the controller, it continues at Step 1.
3. At the end of the transfer, the controller interrupts the processor using the DMA_DONE interrupt for the corresponding channel.

Auto Request

When the controller operates in this mode, it is necessary for the controller to receive a single request to enable it to complete the entire DMA cycle. This allows a large data transfer to occur without significantly increasing the latency for servicing higher priority requests or requiring multiple requests from the processor or peripheral. This mode is very useful for a memory to memory copy application using software DMA requests.

In this mode, the controller can be configured to use either the primary or alternate data structure by programming DMAALTCLR for the corresponding channel. After the channel is enabled, when the controller receives a request, it performs the following operations:

1. The controller performs min(2^{R_POWER},N) transfers for the channel. If the number of transfers remaining is 0, the flow continues at Step 3.
2. A request for the channel is automatically generated. The controller arbitrates. If the channel has the highest priority, the DMA cycle continues at Step 1.
3. At the end of the transfer, the controller interrupts the processor using the DMA_DONE interrupt for that channel.

Ping-Pong

In ping-pong mode, the controller performs a DMA cycle using one of the data structures and then performs a DMA cycle using the other data structure. The controller continues to alternate between using the primary and alternate data structures until it either reads a data structure that is basic/automatic request or the processor disables the channel.

This mode is useful for transferring data using different buffers in the memory. In a typical application, the processor is required to configure both primary and alternate data structures before starting the transfer. As the transfer progresses, the host can subsequently configure primary or alternate control data structures in the interrupt service routine when the corresponding transfer ends.

The DMA controller interrupts the processor using the DMA_DONE interrupt after the completion of transfers associated with each control data structure. The individual transfers using either the primary or the alternate control data structures work exactly the same as a basic DMA transfer.

In this mode, if the DMA request is from software, a request is generated automatically after each arbitration cycle until the completion of the primary or alternate descriptor tasks. The final descriptor must be programmed to use an auto request transfer type. Figure 51 illustrates the process.

If the DMA request is from a peripheral, it must send DMA requests after every data transfer to complete the primary or alternate descriptor tasks, and the final descriptor must be programmed to use a basic transfer type. Figure 52 illustrates the process.

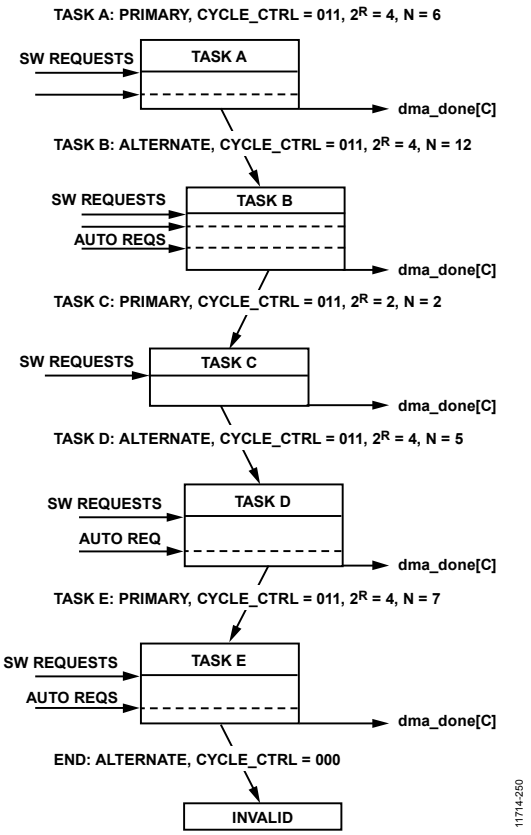


Figure 51. Software Ping-Pong DMA Transfer Process

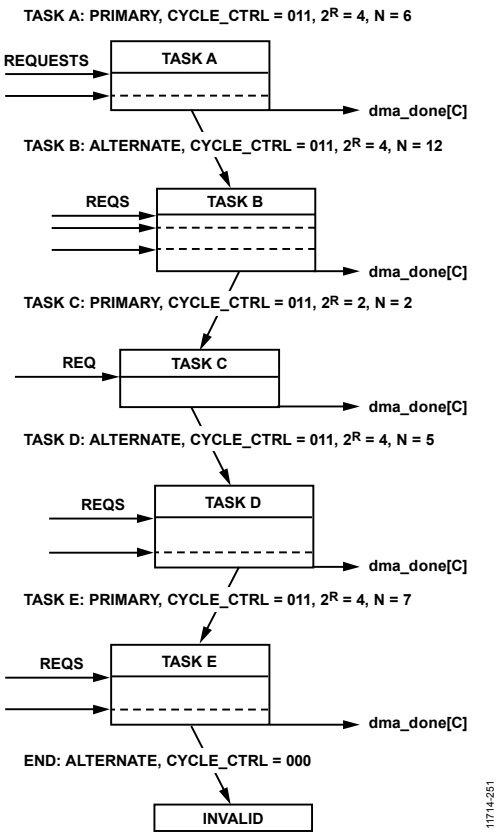


Figure 52. Peripheral Ping-Pong DMA Transfer Process

Memory Scatter-Gather

In memory scatter-gather mode, the DMA controller must be configured to use both primary and alternate data structures. It uses the primary data structure to program the control configuration for alternate data structure. The alternate data structure is used for data transfers using a transfer similar to an auto request DMA transfer. The controller arbitrates after every primary transfer. The controller only needs one request to complete the entire transfer. This mode is used when performing multiple memory to memory copy tasks. The processor can configure all of these tasks together and is not required to intervene in between. The DMA controller interrupts the processor using the DMA_DONE interrupt when the entire scatter gather transaction completes using a basic cycle.

In this mode, the controller receives an initial request and then performs four DMA transfers using the primary data structure to program the control structure of the alternate data structure. After this transfer completes, it starts a DMA cycle using the alternate data structure. After the cycle completes, the controller performs another four DMA transfers using the primary data structure. The controller continues to alternate between using the primary and alternate data structures until either

- The processor configures the alternate data structure for a basic cycle.
- The DMA reads an invalid data structure.

Table 133 lists the fields of the CHNL_CFG memory location for the primary data structure, which must be programmed with constant values for the memory scatter gather mode. Figure 53 further illustrates the memory scatter gather DMA transfer process.

Table 133. CHNL_CFG for Primary Data Structure in Memory Scatter-Gather Mode

Bit(s)	Name	Value	Description
[31:30]	DST_INC	10	Configures the controller to use word increments for the address.
[29:28]	RESERVED	00	Undefined. Write as 0.
[27:26]	SRC_INC	10	Configures the controller to use word increments for the address.
[25:24]	SRC_SIZE	10	Configures the controller to use word transfers.
[23:18]	RESERVED	00	Undefined. Write as 0.
[17:14]	R_POWER	0010	Indicates that the DMA controller is to perform four transfers.
[13:4]	N_MINUS_1	User	Configures the controller to perform N DMA transfers, where N is a multiple of 4.
3	RESERVED	00	Undefined. Write as 0.
[2:0]	CYCLE_CTRL	100	Configures the controller to perform a memory scatter gather DMA cycle.

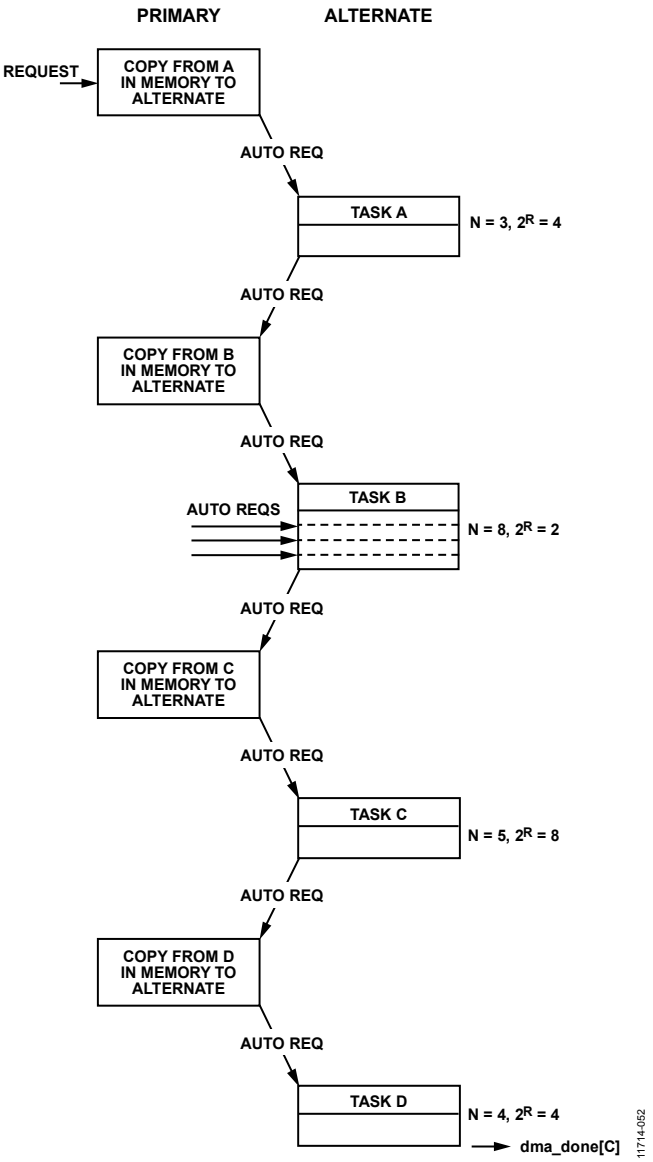


Figure 53. Memory Scatter-Gather Transfer Process

Peripheral Scatter Gather

In peripheral scatter gather mode, the DMA controller must be configured to use both the primary and alternate data structures. The controller uses the primary data structure to program the control structure of the alternate data structure. The alternate data structure is used for actual data transfers, and each transfer takes place using the alternate data structure with a basic DMA transfer. The controller does not arbitrate after every primary transfer. This mode is used when there are multiple peripheral to memory DMA tasks to be performed. The processor can configure all of these tasks at the same time and is not required to intervene in between.

This mode is very similar to memory scatter gather mode except for arbitration and request requirements. The DMA controller interrupts the processor using the DMA_DONE interrupt when the entire scatter gather transaction completes using a basic cycle.

In peripheral scatter gather mode, the controller receives an initial request from a peripheral and then performs four DMA transfers using the primary data structure to program the alternate control data structure. It then immediately starts a DMA cycle using the alternate data structure, without rearbitrating.

After this cycle completes, the controller rearbitrates. If it receives a request with the highest priority from the peripheral, it performs another four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure without rearbitrating. The controller continues to alternate between using the primary and alternate data structures until either

- The processor configures the alternate data structure for a basic cycle.
- The DMA reads an invalid data structure.

Table 134 lists the fields of the CHNL_CFG memory location for the primary data structure, which must be programmed with constant values for the memory scatter gather mode. Figure 54 further illustrates the peripheral scatter gather DMA transfer process.

Table 134. CHNL_CFG for Primary Data Structure in Peripheral Scatter-Gather Mode

Bits	Name	Value	Description
[31:30]	DST_INC	10	Configures the controller to use word increments for the address.
[29:28]	RESERVED	00	Undefined. Write as 0.
[27:26]	SRC_INC	10	Configures the controller to use word increments for the address.
[25:24]	SRC_SIZE	10	Configures the controller to use word transfers.
[23:18]	RESERVED	00	Undefined. Write as 0.
[17:14]	R_POWER	0010	Indicates that the DMA controller is to perform four transfers.
[13:4]	N_MINUS_1	user	Configures the controller to perform N DMA transfers, where N is a multiple of four.
3	RESERVED	00	Undefined. Write as 0.
[2:0]	CYCLE_CTRL	110	Configures the controller to perform a memory scatter gather DMA cycle.

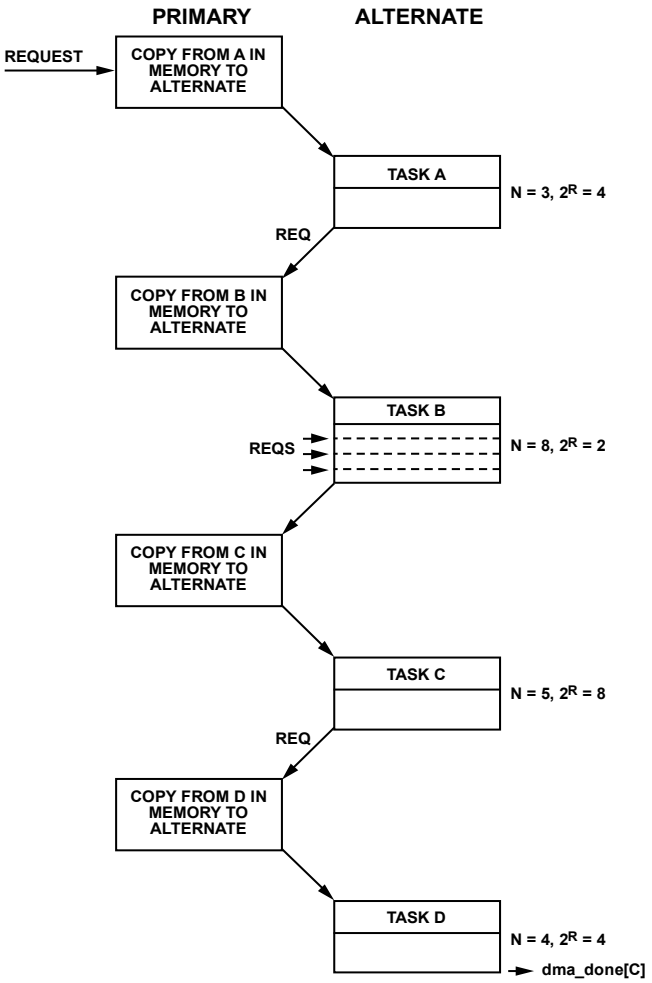


Figure 54. Peripheral Scatter Gather Transfer Process

Error Management

The DMA controller generates an error interrupt to the core whenever a DMA error occurs. A DMA error can occur either because of a bus error or an invalid descriptor fetch.

A bus error can occur while fetching the descriptor or while performing a data transfer. A bus error can occur whenever a read from or write to a reserved address location happens.

When a bus error occurs, the faulty channel is automatically disabled, and the corresponding status bit in DMAERRCHNLCLR is set. If the DMA error is enabled in the NVIC, the error also generates an interrupt. In addition, the CHNL_CFG data structure for the corresponding channel is updated with the latest n_count. The n_count indicates how many successful data transfers happened before the bus error occurred.

When the controller fetches an invalid descriptor (that is, when the cycle control in the configuration control is 0), the faulty channel is automatically disabled, and the corresponding status bit in the DMAINVALIDDESCCLR register is set. If the DMA error is enabled in the NVIC, the error also generates an interrupt.

Address Calculation

The DMA controller calculates the source read address based on the content of the source data end pointer (SRC_END_PTR), the source address increment setting in CHNL_CFG, and the current value of the N_MINUS_1. Similarly, the destination write address is calculated based on the content of the destination data end pointer (DST_END_PTR), the destination address increment setting in CHNL_CFG, and the current value of N_MINUS_1.

$$Source\ Read\ Address = \begin{cases} SRC\_END\_PTR - (N\_MINUS\_1 << (SRC\_INC))\ for\ SRC\_INC = 0, 1, 2 \\ SRC\_END\_PTR\ for\ SRC\_INC = 3 \end{cases}$$

where *N_MINUS_1* is the current count of transfers to be done.

$$\text{Destination Write Address} = \begin{cases} DST\_END\_PTR - (N\_MINUS\_1 << (DST\_INC))\ for\ DST\_INC = 0, 1, 2 \\ DST\_END\_PTR\ for\ DST\_INC = 3 \end{cases}$$

Address Decrement

Address decrement can be enabled to source and destination addresses. Source address decrement can be enabled for channels by setting the appropriate bits in DMASRCADSSET. Similarly, the destination address decrement can be enabled for channels by setting the required bits in DMADSTADSET. The values written into SRC_END_PTR and DST_END_PTR are still used as the addresses for the last transfer as part of the DMA cycle. However, the start address is computed differently according to the address increment scheme for either source read or destination write.

If the source decrement bit is set in DMASRCADSSET for a channel, its source address is computed as follows:

$$Source\ Read\ Address = \begin{cases} SRC\_END\_PTR + (N\_minus\_1 << (SRC\_INC))\ for\ SRC\_INC = 0, 1, 2 \\ SRC\_END\_PTR\ for\ SRC\_INC = 3 \end{cases}$$

If the destination decrement bit is set in DMADSTADSET for a channel, its source address is computed as follows:

$$\text{Destination Write Address} = \begin{cases} DST\_END\_PTR + (N\_minus\_1 << (DST\_INC))\ for\ DST\_INC = 0, 1, 2 \\ DST\_END\_PTR\ for\ DST\_INC = 3 \end{cases}$$

where *N_minus_1* is the current count of transfers to be done.

Byte swap and address decrement must not be used together for any channel. If used together, the DMA data transfer operation is unpredictable.

All the combinations of source and destination decrement are shown with their data movement direction in Figure 55.

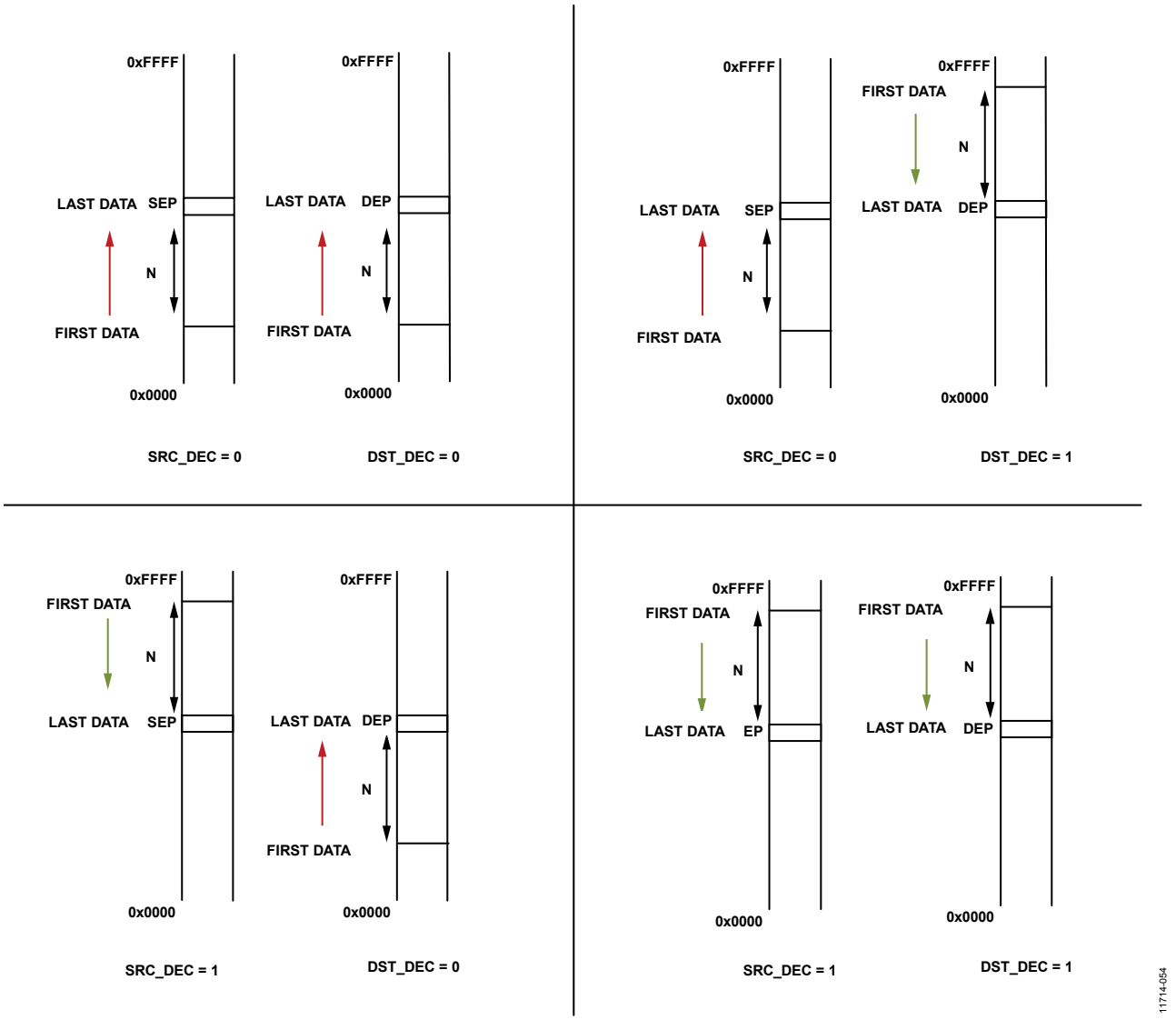


Figure 55. Various Options of Address Decrement

Endian Operation

The DMA controller performs a transfer using a little endian approach by default; however, this default behavior can be changed by setting the corresponding channel bit in the DMABSSET register. The endian operation is referred to as byte swap.

Byte Swap Disabled

Byte swap is disabled by default. In this case, the data transfer is considered to be little endian. Data arriving from a peripheral is placed in sequence, starting from the LSB of a 32-bit word.

For example, if 16 bytes of data arrive at the SPI as 0x01 (start), 0x02, 0x03, 0x04 … 0x0F, 0x10, it is stored by the DMA in memory as follows:

04_03_02_01

08_07_06_05

0C_0B_0A_09

10_0F_0E_0D

Byte Swap Enabled

Byte swap can be enabled on any channel by setting the corresponding bit in the DMABSSET register. By setting this bit, big endian data transfers can occur. Data arriving from the peripheral is placed in sequence, starting from the MSB of a 32-bit word.

For example, if 16 bytes of data arrive at the SPI as 0x01 (start), 0x02, 0x03, 0x04 … 0x0F, 0x10, it is stored by the DMA in memory as follows:

01_02_03_04

05_06_07_08

09_0A_0B_0C

0D_0E_0F_10

Note the following:

- Byte swap happens on 32-bit data boundaries. The transfer size must be a multiple of four.
- Byte swap and address decrement must not be used together for any channel. If used together, the DMA data transfer operation is unpredictable.
- When using byte swap, care must be taken to ensure that the source data address is constant for the full data transfer. For example, SPI where the data source is always from the same location (a FIFO).
- Byte swap functionality is independent of DMA transfer size and can be 8-, 16-, or 32-bit.

DMA Channel Enable/Disable

Before issuing a DMA request, the DMA channel must be enabled. Otherwise, the DMA request for the corresponding channel is driven as a DMA_DONE interrupt. Any DMA channel can be enabled by writing the corresponding bit in the DMAENSET register. The DMA controller disables the channel when the corresponding DMA_DONE interrupt is generated. However, the user can also disable any channel by writing to the corresponding bit in the DMAENCLR register. Whenever the user disables a channel, the following occurs, based on the current state of the DMA controller:

- If the user disables the channel and there is no request pending for that channel, it is disabled immediately.
- If the user disables the channel that is not being serviced, but its request is posted, its pending request is cleared and the channel is disabled immediately.
- If the user disables a channel that has been selected after arbitration but is yet to start transfers, the controller completes the arbitration cycle and then disables the channel.
- If the user disables the channel when it is being serviced, the controller completes the current arbitration cycle and then disables the channel.

DMA Master Enable

- The MENABLE bit in the DMACFG register acts as a soft reset to the DMA controller. Any activity in the DMA controller can be performed only when this bit is set to 1.
- Clearing this bit to 0 clears all cached descriptors within the controller and resets the controller. Table 135 shows the various register values when the DMA controller is reenabled.

Power-Down Considerations

When powering down the chip into hibernate mode, the recommendation is to gracefully finish all the ongoing DMA transfers and then go into hibernation.

However, if the user decides to hibernate as early as possible (current data transfers are ignored), the DMA controller must be disabled by clearing Bit 0 in the DMA configuration register (DMACFG, Address 0x40010004) before going into hibernation.

After hibernation (or POR), the DMA must be enabled again by setting Bit 0 in the DMA configuration register (DMACFG, Address 0x40010004).

DMA CONTROLLER MEMORY MAPPED REGISTERS

DMA Controller Register Map

Table 135. DMA Controller Register Summary

Address	Name	Description	Reset	Reenable Value	RW
0x40010000	DMASTA	DMA status	0x000F0000	PRE_DISABLE_VAL	R
0x40010004	DMACFG	DMA configuration	0x00000000	PRE_DISABLE_VAL	W
0x40010008	DMAPDBPTR	DMA channel primary control data base pointer	0x00000000	PRE_DISABLE_VAL	RW
0x4001000C	DMAADBPTR	DMA channel alternate control data base pointer	0x00000100	PRE_DISABLE_VAL	R
0x40010014	DMASWREQ	DMA channel software request	0x00000000	0x00000000	W
0x40010020	DMARMSKSET	DMA channel request mask set	0x00000000	PRE_DISABLE_VAL	RW
0x40010024	DMARMSKCLR	DMA channel request mask clear	0x00000000	PRE_DISABLE_VAL	W
0x40010028	DMAENSET	DMA channel enable set	0x00000000	0x00000000	RW
0x4001002C	DMAENCLR	DMA channel enable clear	0x00000000	0x00000000	W
0x40010030	DMAALTSET	DMA channel primary-alternate set	0x00000000	0x00000000	RW
0x40010034	DMAALTCLR	DMA channel primary-alternate clear	0x00000000	0x00000000	W
0x40010038	DMAPRISET	DMA channel priority set	0x00000000	PRE_DISABLE_VAL	RW
0x4001003C	DMAPRICLR	DMA channel priority clear	0x00000000	PRE_DISABLE_VAL	W
0x40010048	DMAERRCHNLCLR	DMA per channel error clear	0x00000000	0x00000000	RW
0x4001004C	DMAERRCLR	DMA bus error clear	0x00000000	0x00000000	RW
0x40010050	DMAINVALIDDESCCLR	DMA per channel invalid descriptor clear	0x00000000	0x00000000	RW
0x40010800	DMABSSET	DMA channel bytes swap enable set	0x00000000	PRE_DISABLE_VAL	RW
0x40010804	DMABSCLR	DMA channel bytes swap enable clear	0x00000000	PRE_DISABLE_VAL	W
0x40010810	DMASRCADSSET	DMA channel source address decrement enable set	0x00000000	PRE_DISABLE_VAL	RW
0x40010814	DMASRCADCLR	DMA channel source address decrement enable clear	0x00000000	PRE_DISABLE_VAL	W
0x40010818	DMADSTADSET	DMA channel destination address decrement enable set	0x00000000	PRE_DISABLE_VAL	RW
0x4001081C	DMADSTADCLR	DMA channel destination address decrement enable clear	0x00000000	PRE_DISABLE_VAL	W
0x40010FE0	DMAREVID	DMA controller revision ID	0x00000002	0x00000002	R

DMA Status Register

Address: 0x40010000, Reset: 0x000F0000, Name: DMASTA

Table 136. Bit Descriptions for DMASTA

Bits	Bit Name	Description	Reset	Access
[31:21]	RESERVED	Reserved.	0x0	R
[20:16]	CHNLSM1	Number of available DMA channels minus 1. Number of available DMA channels minus one. With eight channels available, the register reads back 0x07.	0xF	R
[14:8]	RESERVED	Reserved. Undefined.	0x0	R
[7:4]	STATE	Current state of DMA controller. Current state of the DMA control state machine. Provides insight into the operation performed by the DMA at the time this register is read. 0000: idle. 0001: reading channel controller data. 0010: reading source data end pointer. 0011: reading destination end pointer. 0100: reading source data. 0101: writing destination data. 0110: waiting for DMA request to clear. 0111: writing channel controller data. 1000: stalled. 1001: done. 1010: peripheral scatter-gather transition. 1011: undefined. … 1111: undefined.	0x0	R
[3:1]	RESERVED	Reserved. Undefined.	0x0	R

Bits	Bit Name	Description	Reset	Access
0	MENABLE	Enable status of the controller. 0: controller is disabled. 1: controller is enabled.	0x0	R

DMA Configuration Register

Address: 0x40010004, Reset: 0x00000000, Name: DMACFG

Table 137. Bit Descriptions for DMACFG

Bits	Bit Name	Description	Reset	Access
[31:1]	RESERVED	Reserved. Undefined.	0x0	W
0	MENABLE	Controller enable. 0: disable controller. 1: enable controller.	0x0	W

DMA Channel Primary Control Data Base Pointer Register

Address: 0x40010008, Reset: 0x00000000, Name: DMAPDBPTR

The DMAPDBPTR register must be programmed to point to the primary channel control base pointer in the system memory. The amount of system memory that must be assigned to the DMA controller depends on the number of DMA channels used and whether the alternate channel control data structure is used. This register cannot be read when the DMA controller is in the reset state.

Table 138. Bit Descriptions for DMAPDBPTR

Bits	Bit Name	Description	Reset	Access
[31:0]	CTRLBASEPTR	Pointer to the base address of the primary data structure. 5 + log(2) M LSBs are reserved and must be written 0. M is number of channels.	0x0	RW

DMA Channel Alternate Control Data Base Pointer Register

Address: 0x4001000C, Reset: 0x00000100, Name: DMAADBPTR

The DMAADBPTR read only register returns the base address of the alternate channel control data structure. This register removes the necessity for application software to calculate the base address of the alternate data structure. This register cannot be read when the DMA controller is in the reset state.

Table 139. Bit Descriptions for DMAADBPTR

Bits	Bit Name	Description	Reset	Access
[31:0]	ALTCBPTR	Base address of the alternate data structure.	0x100	R

DMA Channel Software Request Register

Address: 0x40010014, Reset: 0x00000000, Name: DMASWREQ

The DMASWREQ register enables the generation of software DMA request. Each bit of the register represents the corresponding channel number in the DMA controller. M is the number of DMA channels

Table 140. Bit Descriptions for DMASWREQ

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	W
[15:0]	CHSWREQ	Generate software request. Set the appropriate bit to generate a software DMA request on the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written, Bit C = 0 does not create a DMA request for Channel C. Bit C = 1 generates a DMA request for Channel C. These bits are automatically cleared by the hardware after the corresponding software request completes.	0x0	W

DMA Channel Request Mask Set Register

Address: 0x40010020, Reset: 0x00000000, Name: DMARMSKSET

Table 141. Bit Descriptions for DMARMSKSET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Reserved, reads back 0.	0x0	R
[15:0]	CHREQMSET	Mask requests from DMA channels. This register disables DMA requests from peripherals. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to mask the request from the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read Bit C = 0, requests are enabled for Channel C. For Bit C = 1, requests are disabled for Channel C. When written as Bit C = 0 there is no effect. Use the DMARMSKCLR register to enable DMA requests. Bit C = 1 disables the peripheral associated with Channel C from generating DMA requests.	0x0	RW

DMA Channel Request Mask Clear Register

Address: 0x40010024, Reset: 0x00000000, Name: DMARMSKCLR

Table 142. Bit Descriptions for DMARMSKCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	CHREQMCLR	Clear REQ_MASK_SET bits in DMARMSKSET. This register enables DMA requests from peripherals by clearing the mask set in DMARMSKSET register. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to clear the corresponding REQ_MASK_SET bit in DMARMSKSET register. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0, there is no effect. Use the DMARMSKSET register to disable DMA requests. Bit C = 1 enables peripheral associated with Channel C to generate DMA requests.	0x0	W

DMA Channel Enable Set Register

Address: 0x40010028, Reset: 0x00000000, Name: DMAENSET

Table 143. Bit Descriptions for DMAENSET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	CHENSET	Enable DMA channels. This register allows for the enabling of DMA channels. Reading the register returns the enable status of the channels. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to enable the corresponding channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read as Bit C = 0, Channel C is disabled. For Bit C = 1, Channel C is enabled. When written as Bit C = 0, there is no effect. Use the DMAENCLR register to disable the channel. Bit C = 1 enables Channel C.	0x0	RW

DMA Channel Enable Clear Register

Address: 0x4001002C, Reset: 0x00000000, Name: DMAENCLR

Table 144. Bit Descriptions for DMAENCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHENCLR	Disable DMA channels. This register allows for the disabling of DMA channels. This register is write only. Each bit of the register represents the corresponding channel number in the DMA controller. Note that the controller disables a channel automatically, by setting the appropriate bit, when it completes the DMA cycle. Set the appropriate bit to disable the corresponding channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0 there is no effect. Use the DMAENSET register to enable the channel. Bit C = 1 disables Channel C.	0x0	W

DMA Channel Primary-Alternate Set Register

Address: 0x40010030, Reset: 0x00000000, Name: DMAALTSET

The DMAALTSET register enables the user to configure the appropriate DMA channel to use the alternate control data structure. Reading the register returns the status of which data structure is in use for the corresponding DMA channel. Each bit of the register represents the corresponding channel number in the DMA controller.

Note that the DMA controller sets/clears these bits automatically as necessary for ping-pong, memory scatter-gather, and peripheral scatter-gather transfers.

Table 145. Bit Descriptions for DMAALTSET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHPRIALTSET	Control structure status/select alternate structure. Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read as Bit C = 0, DMA Channel C uses the primary data structure. For Bit C = 1, DMA Channel C uses the alternate data structure. When written as Bit C = 0, there is no effect. Use the DMAALTCLR register to set Bit C to 0. Bit C = 1 selects the alternate data structure for Channel C.	0x0	RW

DMA Channel Primary-Alternate Clear Register

Address: 0x40010034, Reset: 0x00000000, Name: DMAALTCLR

The DMAALTCLR write only register enables the user to configure the appropriate DMA channel to use the primary control data structure. Each bit of the register represents the corresponding channel number in the DMA controller.

Note that the DMA controller sets/clears these bits automatically as necessary for ping-pong, memory scatter-gather, and peripheral scatter-gather transfers.

Table 146. Bit Descriptions for DMAALTCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHPRIALTCLR	Select primary data structure. Set the appropriate bit to select the primary data structure for the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0, there is no effect. Use the DMAALTSET register to select the alternate data structure. Bit C = 1, selects the primary data structure for Channel C.	0x0	W

DMA Channel Priority Set Register

Address: 0x40010038, Reset: 0x00000000, Name: DMAPRISET

Table 147. Bit Descriptions for DMAPRISET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHPRISET	Configure channel for high priority. This register enables the user to configure a DMA channel to use the high priority level. Reading the register returns the status of the channel priority mask. Each bit of the register represents the corresponding channel number in the DMA controller. This bit returns the channel priority mask status, or sets the channel priority to high. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read as Bit C = 0, DMA Channel C uses the default priority level. For Bit C = 1, DMA Channel C uses a high priority level. When written as Bit C = 0, there is no effect. Use the DMAPRICLR register to set Channel C to the default priority level. For Bit C = 1, Channel C uses the high priority level.	0x0	W

DMA Channel Priority Clear Register

Address: 0x4001003C, Reset: 0x00000000, Name: DMAPRICLR

Table 148. Bit Descriptions for DMAPRICLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHPRICLR	Configure channel for default priority level. The DMAPRICLR write only register enables the user to configure a DMA channel to use the default priority level. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to select the default priority level for the specified DMA channel. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0, there is no effect. Use the DMAPRISET register to set Channel C to the high priority level. For Bit C = 1, Channel C uses the default priority level.	0x0	W

DMA Per Channel Error Clear Register

Address: 0x40010048, Reset: 0x00000000, Name: DMAERRCHNLCLR

Table 149. Bit Descriptions for DMAERRCHNLCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	CHNL_ERR_CLR	Per channel bus error status/per channel bus error clear. This register is used to read and clear the per channel DMA bus error status. The error status is set if the controller encountered a bus error while performing a transfer. If a bus error occurs on a channel, that channel is automatically disabled by the controller. The other channels are unaffected. Write one to clear bits. When read as 0, no bus error occurred. When read as 1, a bus error control is pending. When written as 0, there is no effect. When written as 1 the bit is cleared.	0x0	RW1C

DMA Bus Error Clear Register

Address: 0x4001004C, Reset: 0x00000000, Name: DMAERRCLR

Table 150. Bit Descriptions for DMAERRCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	ERRCLR	Bus error status. This register is used to read and clear the DMA bus error status. The error status is set if the controller encountered a bus error while performing a transfer or when it reads an invalid descriptor (whose cycle control is 3'b000). If a bus error occurs or invalid cycle control is read on a channel, that channel is automatically disabled by the controller. The other channels are unaffected. Write one to clear bits. When read as 0, no bus error/invalid cycle control occurred. When read as 1, a bus error/invalid cycle control is pending. When written as 0, there is no effect. When written as 1, the bit is cleared.	0x0	RW1C

DMA Per Channel Invalid Descriptor Clear Register

Address: 0x40010050, Reset: 0x00000000, Name: DMAINVALIDDESCCLR

Table 151. Bit Descriptions for DMAINVALIDDESCCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	CHNL_INVALID_CLR	Per channel invalid descriptor status/per channel invalid descriptor status clear. This register is used to read and clear the per channel DMA invalid descriptor status. The per channel invalid descriptor status is set if the controller reads an invalid descriptor (whose cycle control is 3'b000). If it reads invalid cycle control for a channel, that channel is automatically disabled by the controller. The other channels are unaffected. Write one to clear bits. When read as 0, no invalid cycle control occurred. When read as 1, an invalid cycle control is pending. When written as 0, there is no effect. When written as 1, the bit is cleared.	0x0	RW1C

DMA Channel Bytes Swap Enable Set Register

Address: 0x40010800, Reset: 0x00000000, Name: DMABSSET

Table 152. Bit Descriptions for DMABSSET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHBSWAPSET	Byte swap status. This register is used to configure a DMA channel to use byte. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read as Bit C = 0, Channel C byte swap is disabled. For Bit C = 1, Channel C byte swap is enabled. When written as Bit C = 0, there is no effect. Use the DMABSCLR register to disable byte swap on Channel C. Bit C = 1 enables byte swap on Channel C.	0x0	RW

DMA Channel Bytes Swap Enable Clear Register

Address: 0x40010804, Reset: 0x00000000, Name: DMABSCLR

Table 153. Bit Descriptions for DMABSCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHBSWAPCLR	Disable byte swap. The DMABSCLR write only register enables the user to configure a DMA channel to not use byte swapping and use the default operation. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0, there is no effect. Use the DMABSSET register to enable byte swap on Channel C. Bit C = 1 disables byte swap on Channel C.	0x0	W

DMA Channel Source Address Decrement Enable Set Register

Address: 0x40010810, Reset: 0x00000000, Name: DMASRCADSSET

Table 154. Bit Descriptions for DMASRCADSSET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHSRCADRDECSET	Source address decrement status/configure source address decrement. The DMASRCADSET register is used to configure the source address of a DMA channel to decrement the address instead of incrementing the address after each access. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read as Bit C = 0, the Channel C source address decrement is disabled. For Bit C = 1, the Channel C source address decrement is enabled. When Written as Bit C = 0, there is no effect. Use the DMAADCLR register to disable source address decrement on Channel C. Bit C = 1 enables source address decrement on Channel C.	0x0	RW

DMA Channel Source Address Decrement Enable Clear Register

Address: 0x40010814, Reset: 0x00000000, Name: DMASRCADCLR

Table 155. Bit Descriptions for DMASRCADCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHADRDECCLR	Disable source address decrement. The DMASRCADCLR write only register enables the user to configure a DMA channel to use the default source address in increment mode. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0, there is no effect. Use the DMASRCADSET register to enable source address decrement on Channel C. Bit C = 1, disables source address decrement on Channel C.	0x0	W

DMA Channel Destination Address Decrement Enable Set Register

Address: 0x40010818, Reset: 0x00000000, Name: DMADSTADSET

Table 156. Bit Descriptions for DMADSTADSET

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHDSTADRDECSET	Destination address decrement status/configure destination address decrement. The DMADSTADSET register is used to configure the destination address of a DMA channel to decrement the address instead of incrementing the address after each access. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When read as Bit C = 0, Channel C destination address decrement is disabled. For Bit C = 1, Channel C destination address decrement is enabled. When written as Bit C = 0, there is no effect. Use the DMADSTADCLR register to disable destination address decrement on Channel C. Bit C = 1, enables destination address decrement on Channel C.	0x0	RW

DMA Channel Destination Address Decrement Enable Clear Register

Address: 0x4001081C, Reset: 0x00000000, Name: DMADSTADCLR

Table 157. Bit Descriptions for DMADSTADCLR

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved. Undefined.	0x0	R
[15:0]	CHADRDECCLR	Disable destination address decrement. The DMAADDSTCLR write only register enables the user to configure a DMA channel to use the default destination address in increment mode. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0 and so on, such that Bit M-1 corresponds to DMA Channel M-1. When written as Bit C = 0, there is no effect. Use the DMAADSET register to enable destination address decrement on Channel C. Bit C = 1 disables destination address decrement on Channel C.	0x0	W

DMA Controller Revision ID Register

Address: 0x40010FE0, Reset: 0x00000002, Name: DMAREVID

Table 158. Bit Descriptions for DMAREVID

Bits	Bit Name	Description	Reset	Access
[31:8]	RESERVED	Reserved. Undefined.	0x0	R
[7:0]	DMAREVID	DMA controller revision ID.	0x2	R

FLASH CONTROLLER

FEATURES

The ADuCM350 platform includes 384 kB of embedded flash memory, accessed using the flash controller. The flash controller is connected to the bus matrix as a slave device for core and DMA access, as well as the 32-bit APB bus for MMR access.

The flash controller supports 384 kB of user space and 2 kB of information space. The 384 kB flash memory comprises one 256 kB flash array and one 128 kB flash array. The 256 kB flash memory array and 128 kB flash array are controlled by two separate flash controllers with separate register controls. Parallel command operations and flash write are possible with the two flash memories.

Commands supported include the following:

- Mass erase and page erase.
- Generation of signatures for single or multiple pages.
- Command abort.

Flash protection includes the following:

- Write protection for user space.
- Ability to lock serial wire interface for read protection.

Flash integrity includes the following:

- Automatic signature check of information space upon reset.
- User signature for application code.
- Parity checking on a per access basis.
- A 128 kB failure rate of less than 0.1 ppm.
- 20,000 cycle endurance with 20 ms erase, 20 µs program.
- 100 year data retention at room temperature.

FLASH MEMORY ORGANIZATION

The flash memory controllers support 128 kB of user flash and 2 kB of information space. Information space is memory mapped above the user flash space, as shown in Figure 56.

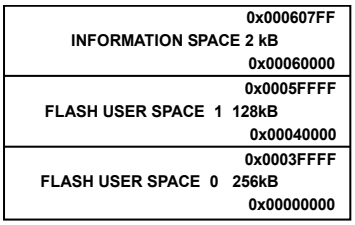


Figure 56. Information and User Space Memory Map

The protection features are common for Flash User Space 0 and Flash User Space 1.

Flash User Space 1

As shown in Figure 57, the top 20 bytes of User Space 1 are reserved for the signature in forward direction, the signature in reverse direction, the user write protection, and the user failure analysis key (FEEFAKEY). Parity Array 1 is applicable only if parity is enabled by setting Bit 0 of the FEE1PARCTL register. The parity feature is discussed in the Flash Integrity, Parity Feature for User Space 1 section. The reserved space from 0x0005_FF80 to 0x0005_FFE8 can be used for user code.

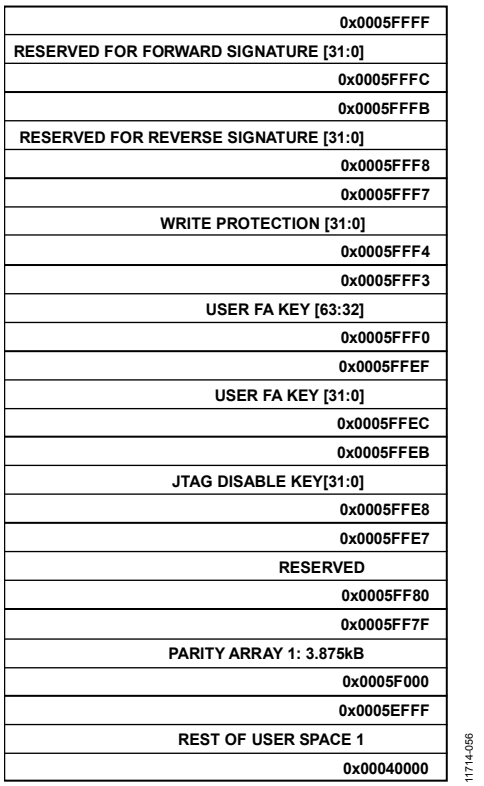


Figure 57. Flash User Space 1 Memory Map

If a user tries to read or write from/to a portion of memory that is not available, a bus error is returned.

Flash User Space 0

Parity Array 0, as shown in Figure 58, is applicable only if parity is enabled by setting Bit 0 of the FEE0PARCTL register. The parity feature is discussed in the Flash Integrity, Parity Feature for User Space 0 section.

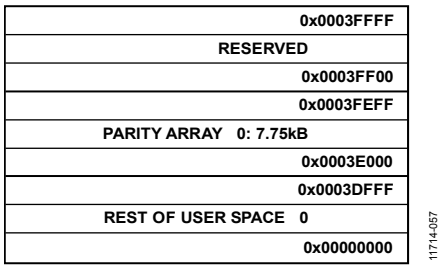


Figure 58. Flash User Space 0 Memory Map

Address Used in Registers for Flash Controller 0

Figure 59. Flash User Space 0 Offset Memory Map

For the following registers, only the offset address is written or available as read back:

- FEE0ADR0x (FEE0ADR0L, FEE0ADR0H)
- FEE0ADR1x (FEE0ADR1L, FEE0ADR1H)
- FEE0ADRAx (FEE0ADRAL, FEE0ADRAH)
- FEE0PARADRx (FEE0PARADRL, FEE0PARADRH)

Full memory mapped addresses, as in Figure 56, are not applicable for these registers. They are applicable for accesses via the AHB only.

Address Used in Registers for Flash Controller 1

Figure 60. Flash User Space 1 Offset Memory Map

For the following registers, only the offset address is written or available as read back, as per Figure 60:

- FEE1ADR0x (FEE1ADR0L, FEE1ADR0H)
- FEE1ADR1x (FEE1ADR1L, FEE1ADR1H)
- FEE1ADRAx (FEE1ADRAL, FEE1ADRAH)
- FEE1PARADRx (FEE1PARADRL, FEE1PARADRH)

Full memory mapped addresses, as in Figure 56, are not applicable for these registers. They are applicable for accesses via the AHB only.

Writing to Flash Memory

Flash is written directly using an AHB write. Only 32-bit writes are supported; 16- and 8-bit writes are not supported. Burst writes to flash are supported. Because only 0s are written to flash, masking can be used to write individual bits or bytes, if necessary. A single location can only be written twice without an erase. If the addressed location is write protected, the controller responds with a bus fault exception.

If a write is followed immediately by a second write to the next location in flash, and this is in the same row as the previous write, the flash write time is faster because the controller does not need to change the row address. This is reflected in the Performance, Command Duration section.

Before performing a write to Flash User Space 0, the FEE0STA register must be read and checked to ensure that no other commands/writes are being executed. Similarly, before performing a write to Flash User Space 1, the FEE1STA register must be read and checked to ensure that no other commands/writes are being executed.

Flash Protection

There are three types of protection implemented: key protected registers, read protection, and write protection.

Key Protected Registers for User Space 0

Some of the flash controller registers are key protected to avoid accidental writes to these registers.

The user key is 0xF123F456. It is entered via the 16-bit key register (FEE0KEY): 0xF456 first followed by 0xF123. This key must be entered to run certain user commands. Commands are written to the FEE0CMD register. When the command completes, the key is cleared automatically.

Key Protected Registers for User Space 1

Some of the flash controller registers are key protected to avoid accidental writes to these registers.

The user key is 0xF123F456. It is entered via the 16-bit key register (FEE1KEY): 0xF456 first followed by 0xF123. This key must be entered to run certain user commands, to write to certain locations in flash, or to enable write access to the setup register. The key remains asserted after being entered, unless a command is written to the FEE1CMD register. When the command completes, the key is cleared automatically. If this key is entered to enable write access to the setup register (FEECON1) or to enable writes to certain locations in flash, it must be cleared by user code afterwards. To clear it, write any 16-bit value to the key register (FEE1KEY).

User Read Protection

User space read protection is provided by disabling the serial wire access. A user can disable the serial wire access by writing 0 to DBG in the flash FEECON1 register. This may be useful if a user wants to block all access to the device except via a downloader, which may have its own security (maybe password protected). It is possible to reenable the serial wire interface via the downloader.

Serial wire access is disabled while the kernel is running. When the kernel has completed, it enables the serial wire access by writing 1 to DBG in the flash FEECON1 register. User code can then clear DBG immediately.

As a feature, JTAG/serial wire access can be user controlled, preventing debug access and viewing of user code.

Upon power-up, JTAG/serial wire access is disabled if a control key has been programmed by the user in the reserved space of user flash (Address 0x5FFE8). Otherwise, the JTAG/serial wire access is enabled. The value of the control key is 0x16032010. Additionally, JTAG/serial wire access can be enabled/disabled in the application code by writing to the FEECON1 register.

User Write Protection

User write protection is provided to prevent accidental writes to pages in user space and to protect blocks of user code when downloading extra code to flash. If a write or erase of a protected location is detected over the DCODE bus, the controller responds with an AHB error.

The write protection is stored at the top of Flash User Space 1. The top eight bytes are reserved for a signature, and the next four bytes are for write protection. The write protection is uploaded by the flash controller into the local registers after reset. After uploading, the 32 write protection bits can be read via the memory mapped registers, FEEPROx[31:0] (FEEPROH[15:0], the upper 16 bits, and FEEPROL[15:0], the lower 16 bits). To write to the write protection bits, a user must first write 0xF456 followed by 0xF123 to the key register (FEE1KEY).

The sequence for programming write protection[31:0] follows (see Figure 57):

1. Write 0xF456 followed by 0xF123 to the key register, FEE1KEY.
2. Write the required write protection directly to flash. Write 0 to enable protection. The write protection address is 0x5FFF4 for 384 kB of flash.
3. Verify that the write completed successfully by polling the status register or by enabling a write complete interrupt.
4. Reset the device, and the write protection is uploaded from information space and activated by the flash controller.

If the write protection in flash has not been programmed, that is, 0xFFFFFFFF is uploaded from flash upon power-up, the FEEPROH and FEEPROL registers can be written to directly from the APB interface. This allows a user to verify write protection before committing it to flash.

If the write protection in flash has been programmed, the MSB of write protection must be programmed to 0 to prevent write protection erasing.

For write protection, Flash User Space 0 (256 kB) is split into 16 blocks and Flash User Space 1 (128 kB) is split into 16 blocks.

User Space 0 is protected by Bits[15:0] of the write protection bits; each bit corresponds to a block size of 16 kB in User Space 0.

User Space 1 is protected by Bits[31:16] of the write protection bits; each bit corresponds to a block size of 8 kB in User Space 1.

After the write protection word has been written, it cannot be rewritten without a full erase (mass erase) of User Space 1 and a full erase (mass erase) of User Space 0.

If an attempt is made to write to the write protection word in flash without setting FEE1KEY first, a bus error is generated.

Failure Analysis Key

It may be necessary to perform failure analysis on devices that are returned by a user but read protection is enabled. A method is provided to allow failure analysis of protected memory; this is a user failure analysis allow key.

This key is a 64-bit key that is stored at the top of user space in flash. It is used to gain access to user code if the serial wire interface is locked. It is the user's responsibility to program this key to a value. The key must be given to Analog Devices to enable access to the user code.

Flash Integrity, Signature Feature

The signature is used to check the integrity of the flash device. Software can call a signature check command occasionally or whenever a new block of code is about to be executed. The signature is a 32-bit CRC with the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (IEEE 802.3). See the CRC Accelerator section for a description of the CRC algorithm used for signal generation.

The sign command can be used to generate a signature and check the signature of a block of code, where a block can be a single page or multiple pages. A 32-bit LFSR is used to generate the signature. The hardware assumes that the signatures for a block are stored within the upper eight bytes of the most significant page of a block; therefore, these eight bytes are not included when generating the signature. The following sequence must be followed to generate a forward signature.

1. Write the start address (lower address) of the block to the PageAddr0 register (FEE0ADR0x or FEE1ADR0x).
2. Write the end address (higher address) of the block to the PageAddr1 register (FEE0ADR1x or FEE1ADR1x).
3. Write the FSIGN command to the command register.
4. When the command has completed, the signature is available in the sign register. Scan is in the forward direction; therefore, the signature is compared with the data stored in the upper 4 bytes of the uppermost page of the block.
5. If the data does not match the signature, a fail status of VerifyErr (FEE0STA[5:4] or FEE1STA[5:4] = 10) is returned in the status register.

The following sequence must be followed to generate a reverse signature.

1. Write the end address (lower address) of the block to the PageAddr0 register.
2. Write the start address (higher address) of the block to the PageAddr1 register.
3. Write the RSIGN command to the command register.
4. When the command has completed, the signature is available in the sign register. Scan is in the reverse direction; therefore, the signature is compared with the data stored in the 4 bytes that precede the upper 4 bytes of the uppermost page of the block.
5. If the data does not match the signature, a fail status of VerifyErr is returned in the status register.

While the signature is being computed, all other accesses to flash are stalled. For a 256 kB block, that is, 64k reads.

Note that PageAddr0/PageAddr1 addresses are byte addresses, but only pages must be identified; that is, the lower 11 bits are ignored by the hardware.

The FSIGN and RSIGN commands and Page Addr 0/Page Addr 1 and status registers are separately available for Flash User Space 0 and Flash User Space 1. The sign command can be executed in parallel for User Space 0 and User Space 1.

Relevant registers for running the sign command in User Space 0 include the following:

- FEE0CMD: command register
- FEE0ADR0H (upper 16 bits of the FEE0ADR0x[31:0] word), FEE0ADR0L (lower 16 bits of the FEE0ADR0x[31:0] word: Page Addr0 register
- FEE0ADR1H (upper 16 bits of the FEE0ADR1x[31:0] word), FEE0ADR1L (lower 16 bits of the FEE0ADR1x[31:0] word: Page Addr1 register
- FEE0STA: status register

Relevant registers for running the sign command in User Space1 include the following:

- FEE1CMD: command register
- FEE1ADR0H (upper 16 bits of the FEE1ADR0x[31:0] word), FEE1ADR0L (lower 16 bits of the FEE1ADR0x[31:0] word: Page Addr0 register
- FEE1ADR1H (upper 16 bits of the FEE1ADR1x[31:0] word), FEE1ADR1L (lower 16 bits of the FEE1ADR1x[31:0] word: Page Addr1 register
- FEE1STA: status register

Integrity of Information Space

The hardware automatically checks the integrity of the information space after reset. In the event of a failure, Bit 6 (SIGNERR) of the FEE1STA register is set, and user code does not run. If the serial wire interface is enabled, this bit can only be read via a serial wire read.

FLASH INTEGRITY, PARITY FEATURE FOR USER SPACE 0

All code accesses by the core have the option of parity checking. At the time of flash image generation, parity values are generated on a per 32-bit accesses basis for relevant locations within the 248 kB region, from 0x00000000 to 0x0003DFFC. Parity values are generated at the time the flash image is made; generation is not supported as a hardware function. These parity values (one bit for each 32-bit access) are then stored in a 7.75 kB region of flash, from 0x0003E000 to 0x0003FEFF. The parity region is shown in Figure 58. As ICODE and DCODE flash accesses occur, a parity bit (even) is generated based on each 32-bit access value. This generated parity value is then compared to the stored parity value. The code access address is used to generate the appropriate index to the corresponding stored parity address and bit position. The index can be derived from the following equation:

$$PARITY\_ADDRESS = (HEXADDRESS >> 7) \times (0x4) + 0x0003E000$$

$$PARITY\_BIT = HEXADDRESS[6:2]$$

Note that HEXADDRESS[1:0] always equals 00.

The parity bit is not stored with code data because this requires 33-bit memory.

Parity checking occurs only if the parity enable (PAREN) bit in the FEE0PARCTL register is set. If no parity checking is required, the PAREN bit must be cleared, and the parity storage region (0x0003E000 to 0x0003FEFF) can be used for code or data storage with no restrictions.

If an access results in a parity error, the parity error (PARERR) bit in FEE0PARSTA is set, and the code address that was responsible for the parity error is captured in the parity error address (FEE0PARADRL, FEE0PARADRH) registers. Optionally, a parity interrupt can be generated on an access parity error if enabled by setting the parity error exception enable (PERREXEN) bit in the FEE0PARCTL register.

FLASH INTEGRITY, PARITY FEATURE FOR USER SPACE 1

All code accesses by the core have the option of parity checking. At the time of flash image generation, parity values are generated on a per 32-bit accesses basis for relevant locations within the 124 kB region, from 0x00040000 to 0x0005EFFF. Parity values are generated at the time the flash image is made; generation is not supported as a hardware function. These parity values (one bit for each 32-bit access) are then stored in a 3.875 kB region of flash, from 0x0005F000 to 0x0005FF7F. The parity region is shown in Figure 57. As ICODE and DCODE flash accesses occur, a parity bit (even) is generated based on each 32-bit access value. This generated parity value is then compared to the stored parity value. The code access address is used to generate the appropriate index to the corresponding stored parity address and bit position. The index can be derived by the following equation:

$$PARITY\_ADDRESS = (HEXADDRESS - 0x00040000 >> 7) \times (0x4) + 0x0005F000$$

$$PARITY\_BIT = HEXADDRESS[6:2]$$

Note that HEXADDRESS[1:0] always equals 00.

The parity bit is not stored with code data because this require 33-bit memory.

Parity checking occurs only if the parity enable (PAREN) bit in the FEE1PARCTL register is set. If no parity checking is required, the PAREN bit must be cleared, and the parity storage region (0x0005F000 to 0x0005FF7F) can be used for code or data storage with no restrictions.

If an access results in a parity error, the parity error (PARERR) bit in the FEE1PARSTA register is set, and the code address that was responsible for the parity error is captured in the parity error address (FEE1PARADRL, FEE1PARADRH) registers. Optionally, a parity interrupt can be generated on an access parity error if enabled by setting the parity error exception enable (PERREXEN) bit in the FEE1PARCTL register.

PARITY ERROR INTERRUPT

When a parity error is detected while fetching instructions or data from flash memory, the interrupt service routine corresponding to the parity error must be executed on the highest priority compared to all other interrupts. Parity error interrupt is generated when the parity error interrupt is enabled and there is a parity error detected either from Flash Controller 0 or from Flash Controller 1. After entering the ISR, another parity interrupt cannot be generated until the current parity interrupt status bit is cleared using write 1 to clear the bit, FEE0PARSTA[0] or FEE1PARSTA[0].

However, during the course of the ISR, the user has the option of clearing any pending parity interrupt and disabling further parity checking. Parity Error Interrupt IRQ Position 60 is the flash parity error interrupt (see the System Interrupts and Exceptions section).

ABORT USING SYSTEM INTERRUPTS

Commands (erase, sign, or mass verify) and writes can be aborted upon receipt of an interrupt. Aborts are also possible by writing an abort command to the command register. However, if flash is being programmed, and the routine controlling the programming is in flash, it is not possible to use the abort command to abort the cycle because instructions cannot be read. Therefore, the ability to abort a cycle on the assertion of any system interrupt is provided. The system IRQ abort enable register is used to enable aborts upon receipt of an interrupt.

When a command or write is aborted via a system interrupt, a CMDDONE or WRDONE status bit is asserted, and CMDRES in the status register indicates aborted.

It is not possible to abort a flash write or erase cycle without waiting for the high voltage in the flash core to discharge first; that is, ensuring a write cycle completes by waiting 6.6 µs for the high voltage to discharge. An erase cycle must finish with 6.6 µs, and a mass erase cycle must finish with 121 µs.

Depending on the state of a write cycle when the abort asserts, the write cycle may or may not complete. If the write or erase cycle did not complete successfully, an aborted fail status can be read in the status register.

If an immediate response to an interrupt is required during an erase or program cycle, the interrupt service routine and the interrupt vector table must be moved to SRAM for the duration of the cycle.

If the DMA engine is set up to write a block of data to flash, an interrupt can be set up to abort the current write; however, the DMA engine starts the next write immediately. The interrupt causing the abort stays asserted; therefore, there are several aborted write cycles, in this case, before the processor can access flash.

When an abort is triggered by an interrupt, all commands are repeatedly aborted until the appropriate system IRQ abort enable register bit is cleared, or the interrupt source is cleared.

Abort operation can be enabled separately for Flash User Space 0 and Flash User Space 1.

Relevant registers for Flash User Space 0 include the FEE0AEN (system IRQ abort enable) register and the FEE0STA (status) register.

Relevant registers for Flash User Space 1 include the FEE1CMD (command) register, the FEE1AEN (system IRQ abort enable) register, and the FEE1STA (status) register.

POWER-DOWN INSTRUCTIONS

Active Mode to Core Sleep Mode

This mode is similar to general-purpose flash. Writes or commands can execute in parallel, and the device can enter sleep mode. However, commands can occur only on the opposite flash from which the current code is being executed.

Active Mode to System Sleep Mode

Same as hibernate mode.

Active Mode to Hibernate Mode

DMA to flash occurs as follows:

- IRQ abort must be enabled to abort all writes in progress and allow entering power-down mode.
- Because of the write, the DCODE bus is busy and no code can execute. Code can execute after the DMA is done or aborted. Use a level interrupt to abort all writes. Pulse interrupt can only abort the current write.
- Level interrupt must be cleared only when both FEE0STA and FEE1STA read back 0 (busy related bits only). Only then are all writes aborted.

DMA from flash occurs as follows:

- ICODE access is available. The DMA can be disabled using code and then go to sleep.

FLASH CRASH

If a flash write or erase is in progress and there is a sudden removal of power without warning, the following cases must be considered:

- Instruction flash program or erase.
- General-purpose program or erase.

The following must be noted:

- At the lowest level, the key array signals are address, data, erase, program, and power. In addition, there are key timing relationships between these signals.
- It is difficult to predict exactly what will happen with these key signals.
- It is also difficult to predict what effect each situation will have on the array.
- A command on one array cannot affect any another array.

Instruction Flash

If power is removed while programming, there are a two possible outcomes.

- Incomplete programming of a location.
- An unintended location is programmed.

Possible actions are as follows:

- Signatures whenever possible.
- Program Page 0 signature last. The kernel does not give control to user code if Page 0 signature command fails.

Protections are the following:

- There is a hardware feature implemented that aborts flash commands (including program and erase) if an interrupt is detected. This feature does not require code execution.
- A HP_LDO power supply monitor interrupt can be used as an abort interrupt source. This is the fastest track from time of detection.

General-Purpose Flash

Interrupt hardware logic to abort a general-purpose flash command is available and is the same as the instruction flash.

- The core is not required to abort a general-purpose flash command.
- Use this feature sparingly.

Possible actions are as follows:

- Use signatures whenever possible.
- Maintain a last valid record indicator and program it last.
- Program half a word within a record to indicate a record's programming has started. Program the other half at completion.

Shut down DMA based flash programming.

- Aborting the current flash program cycle does not disable the programming of the DMA.

PERFORMANCE, COMMAND DURATION

See the ADuCM350 data sheet for more information. All values listed here are for reference purposes only.

Direct single write access (32-bit location): 46 µs (typical).

Mass erase: 21 ms (typical); larger arrays incur longer erase times.

Page erase: 21 ms (typical).

Direct write of a page (512, 32-bit locations): 12.16 ms (typical).

Mass verify for all user space, 256 kB: 4.1 ms (256 kB and 16 MHz HCLK). This is the array data read component; all other time associated with the overhead of data comparison and moving data on and off the chip is not included and must be added based on the user's verification strategy.

Sign for all user space, 256 kB: 8.2 ms (256 kB and 16 MHz HCLK). The value listed here is for a single pass, either in a forward or reverse direction.

FLASH CONTROLLER MEMORY MAPPED REGISTERS

The Flash User Space0 memory array (256 kB) is controlled by Flash Controller0, and the Flash User Space1 memory array (128 kB) is controlled by Flash Controller 1. All the MMRs in the flash controller are duplicated except for the protection related registers. The flash protection related registers are placed in the MMRs of Flash Controller1.

Register Map for Flash Parity 0

Table 159. Flash Parity Register Summary

Address	Name	Description	Reset	RW
0x40018000	FEE0STA	Status	0x0000	R
0x40018004	FEE0CON0	Command control	0x0000	RW
0x40018008	FEE0CMD	Command	0x0000	RW
0x40018010	FEE0ADR0L	Lower page address	0x0000	RW
0x40018014	FEE0ADR0H	Upper page address	0x0000	RW
0x40018018	FEE0ADR1L	Lower page address	0x0000	RW
0x4001801C	FEE0ADR1H	Upper page address	0x0000	RW
0x40018020	FEE0KEY	Key	0x0000	W
0x40018030	FEE0SIGL	Lower halfword of signature	0xXXXX	R
0x40018034	FEE0SIGH	Upper halfword of signature	0xXXXX	R
0x40018048	FEE0ADRAL	Lower halfword of write abort address	0xXXXX	R
0x4001804C	FEE0ADRAH	Upper halfword of write abort address	0xXXXX	R
0x40018050	FEE0PARCTL	Parity control register	0x0000	RW
0x40018054	FEE0PARSTA	Parity status register	0x0000	RW1C
0x40018058	FEE0PARADRL	Parity error address low	0x0000	R
0x4001805C	FEE0PARADRH	Parity error address high	0x0000	R
0x40018078	FEE0AEN0	System IRQ abort enable for Interrupt 15 to Interrupt 0	0x0000	RW
0x4001807C	FEE0AEN1	System IRQ abort enable for Interrupt 31 to Interrupt 16	0x0000	RW
0x40018080	FEE0AEN2	System IRQ abort enable for Interrupt 47 to Interrupt 32	0x0000	RW
0x40018084	FEE0AEN3	System IRQ abort enable for Interrupt 60 to Interrupt 48	0x0000	RW

Status Register

Address: 0x40018000, Reset: 0x0000, Name: FEE0STA

Table 160. Bit Descriptions for FEE0STA

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved.	0x0	R
[5:4]	CMDRES	Command result. These two bits indicate the status of a command on completion or the status of a write via the AHB. If multiple commands are executed or there are multiple writes via the AHB without a read of the status register then the first error encountered is stored. These bits clear to 00 when read. After an erase, the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all Fs, this is the resulting status. If the sign command is executed and the resulting signature does not match the data in four of the upper eight bytes of the upper page in a block then this is the resulting status. 00: successful completion of a command or a write. 01: attempted erase of a protected location. 10: read verify error. 11: indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.	0x0	RC
3	WRDONE	Write complete. This bit asserts when a write via the AHB completes. It clears when read. If there are multiple writes (or a burst write), this status bit asserts after the first long word written and stays asserted until read. If there is a burst write to flash then this bit asserts after every long word written (assuming the bit is cleared by a read after every long word written).	0x0	RC
2	CMDDONE	Command complete. This bit asserts when a command completes. It clears when read. If there are multiple commands, this status bit asserts after the first command completes and stays asserted until read.	0x0	RC
1	WRBUSY	Write busy. This bit is asserted when the flash block is executing a write via the AHB.	0x0	R
0	CMDBUSY	Command busy. This bit is asserted when the flash block is executing any command entered via the command register.	0x0	R

Command Control Register

Address: 0x40018004, Reset: 0x0000, Name: FEE0CON0

Table 161. Bit Descriptions for FEE0CON0

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved. Returns 0 when read.	0x0	R
2	WREN	Write enable. When WREN is 1, it is possible to write to flash. When it is 0, an attempt to write to a flash location results in an AHB error and the write does not take place.	0x0	RW
1	IENERR	Error interrupt enable. If this bit is set then an interrupt is generated when a command or flash write completes with an error status.	0x0	RW
0	IENCMD	Command complete interrupt enable. When set, an interrupt is generated when a command or flash write completes.	0x0	RW

Command Register

Address: 0x40018008, Reset: 0x0000, Name: FEE0CMD

The command register is used to initiate flash operations such as signature, erase and abort. This register can be read at any time. If a command is in progress, the only write to this register is the ABORT command. All commands, including the IDLE command, result in a command complete interrupt (CMDDONE).

Table 162. Bit Descriptions for FEE0CMD

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved. Always returns 0 when read.	0x0	RW
[3:0]	CMD	IDLE: the flash controller takes no action. A command complete interrupt is issued if an idle command is initiated. It is not necessary to issue the IDLE command at the completion of any of the other command operations. ERASEPAGE: write the address of the page to be erased to the PageAddr0 register, then write this code to the FEECMD register and the flash erases the page. When the erase has completed, the flash reads every location in the page to verify all words in the page are erased. If there is a read verify error, it is indicated in the status register. To erase multiple pages, wait until a previous page erase has completed. Check the status and then issue a command to start the next page erase. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register. FSIGN: use this command to generate a forward direction signature for a block of data. The signature is generated on a page by page basis. To generate a signature, the address of the first page (lower address) of the block is entered in the PageAddr0 register, the address of the last page (higher address) is written to the PageAddr1 register, and then this code is written to the CMD register. The signature calculation is in forward direction (that is, it starts from the least significant word (LSW) of lower page address (first page) and works by incrementing the address). When the command is complete, the signature is available for reading in the sign register. The last eight bytes of the higher address page in a block is reserved for storing the reverse and forward signature. The forward signature is placed in the last four bytes. Before entering this command, 0xF456 followed 0xF123 must be written to the key register. RSIGN: use this command to generate a reverse direction signature for a block of data. The signature is generated on a page by page basis. To generate a signature the address of the last page (lower address) of the block is entered in the PageAddr0 register, the address of the first page (higher address) is written to the PageAddr1 register, and then this code is written to the CMD register. The signature calculation is in reverse direction (that is, it starts from the most significant word (MSW) of higher page address (first page) and works by decrementing the address). When the command is complete, the signature is available for reading in the sign register. The last eight bytes of the higher address page in a block are reserved for storing the reverse and forward signature. The reverse signature (four bytes) is placed on four bytes that precede forward signature (last four bytes). Before entering this command, 0xF456 followed by 0xF123 must be written to the key register. For FSIGN and RSIGN commands, the lower address is always placed in the PageAddr0 register and the higher address is placed in the PageAddr1 register. If signature check is on a single page, the corresponding page address is placed in both the PageAddr0 and PageAddr1 registers. MASSERASE: erase all of user space. To enable this operation, 0xF456 followed by 0xF123 must first be written to the key register (this is to prevent accidental erases). When the mass erase is completed, the controller reads every location to verify that all locations are 0xFFFFFFFF. If there is a read verify error this, it is indicated in the status register.	0x0	RW

Bits	Bit Name	Description	Reset	Access
		ABORT: if this command is issued, then any command currently in progress is stopped. The status indicates that the command completed with an error status of CMDABORTED. Note that this is the only command that can be issued while another command is already in progress. This command can also be used to stop a write that may be in progress. If a write is aborted, the address of the written location can be read via the FEEADRA register. While the flash controller is writing one long word another long word write may be in the pipeline from the Cortex-M3 or DMA engine (depending on how the software implements writes). Therefore, both writes may need to be aborted. If a write or erase is aborted, then the flash timing is violated and it is not possible to determine if the write or erase completed successfully. To enable this operation, 0xF456 followed by 0xF123 must first be written to the key register (this is to prevent accidental aborts). 0000: IDLE command. 0001: ERASEPAGE command. 0010: FSIGN command. 0011: RSIGN command. 0100: MASSERASE command. 0101: ABORT command.		

Lower Page Address Register

Address: 0x40018010, Reset: 0x0000, Name: FEE0ADR0L

Table 163. Bit Descriptions for FEE0ADR0L

Bits	Bit Name	Description	Reset	Access
[15:11]	LOW	Lower five bits of page address. Lower five bits of the address of a page in flash. Used by the erase and sign commands.	0x0	RW
[10:0]	RESERVED	Reserved. Page size is 2 kB. These 11 bits are marked as reserved so that byte addresses can be written directly to this register. The lower 11 bits of a byte address are not used because this is a page address. Returns 0x0 if read.	0x0	RW

Upper Page Address Register

Address: 0x40018014, Reset: 0x0000, Name: FEE0ADR0H

Table 164. Bit Descriptions for FEE0ADR0H

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	RW
[1:0]	HIGH	Upper bits of page address. Upper address bits of the page address.	0x0	RW

Lower Page Address Register

Address: 0x40018018, Reset: 0x0000, Name: FEE0ADR1L

Table 165. Bit Descriptions for FEE0ADR1L

Bits	Bit Name	Description	Reset	Access
[15:11]	LOW	Lower five bits of page address. Lower five bits of the address of a page in flash. Used by the erase and sign commands.	0x0	RW
[10:0]	RESERVED	Reserved. Page size is 2 kB. These 11 bits are marked as reserved so that byte addresses can be written directly to this register. The lower 11 bits of a byte address are not used because this is a page address. Returns 0x0 if read.	0x0	RW

Upper Page Address Register

Address: 0x4001801C, Reset: 0x0000, Name: FEE0ADR1H

Table 166. Bit Descriptions for FEE0ADR1H

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	RW
[1:0]	HIGH	Upper bits of page address. Upper address bits of the page address.	0x0	RW

Key Register

Address: 0x40018020, Reset: 0x0000, Name: FEE0KEY

Table 167. Bit Descriptions for FEE0KEY

Bits	Bit Name	Description	Reset	Access
[15:0]	KEY	Key. Enter 0xF456 followed by 0xF123. Returns 0x0 if read.	0x0	W

Lower Halfword of Signature Register

Address: 0x40018030, Reset: 0x000X, Name: FEE0SIGL

Table 168. Bit Descriptions for FEE0SIGL

Bits	Bit Name	Description	Reset	Access
[15:0]	SIGL	Lower halfword of signature. Corresponds to SIGNATURE[15:0].	0xx	RW

Upper Halfword of Signature Register

Address: 0x40018034, Reset: 0x000X, Name: FEE0SIGH

Table 169. Bit Descriptions for FEE0SIGH

Bits	Bit Name	Description	Reset	Access
[15:0]	SIGH	Upper byte of the signature. Corresponds to SIGNATURE[31:16].	0xx	R

Lower Halfword of Write Abort Address Register

Address: 0x40018048, Reset: 0x000X, Name: FEE0ADRAL

Table 170. Bit Descriptions for FEE0ADRAL

Bits	Bit Name	Description	Reset	Access
[15:0]	FEEADRAL	Lower halfword of FEEADRA. If a write is aborted then this contains the address of the location written when the write was aborted. This register has relevant value only if appropriate flags in FEE0STA register are set after a write abort.	0xx	R

Upper Halfword of Write Abort Address Register

Address: 0x4001804C, Reset: 0x000X, Name: FEE0ADRAH

Table 171. Bit Descriptions for FEE0ADRAH

Bits	Bit Name	Description	Reset	Access
[15:0]	FEEADRAH	Upper halfword of FEEADRA.	0xx	R

Parity Control Register

Address: 0x40018050, Reset: 0x0000, Name: FEE0PARCTL

Table 172. Bit Descriptions for FEE0PARCTL

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved. Returns a value of 0 when read.	0x0	R
1	PERREXEN	Parity error interrupt enable. 0: a parity error does not result in a parity interrupt. 1: a parity error results in a parity interrupt.	0x0	RW
0	PAREN	Parity enable. 0: parity checking is disabled. 1: parity checking is enabled for each ICODE and DCODE flash access.	0x0	RW

Parity Status Register

Address: 0x40018054, Reset: 0x0000, Name: FEE0PARSTA

Table 173. Bit Descriptions for FEE0PARSTA

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Returns a value of 0 when read.	0x0	R
0	PARERR	Parity error. 0: no parity error has been detected since last cleared. 1: a parity error has been detected.	0x0	RW1C

Parity Error Address Low Register

Address: 0x40018058, Reset: 0x0000, Name: FEE0PARADRL

Table 174. Bit Descriptions for FEE0PARADRL

Bits	Bit Name	Description	Reset	Access
[15:0]	ADDRL	Parity error address low. Indicates the lower 16 bits of the address are flagged as having an access parity error.	0x0	R

Parity Error Address High Register

Address: 0x4001805C, Reset: 0x0000, Name: FEE0PARADRH

Table 175. Bit Descriptions for FEE0PARADRH

Bits	Bit Name	Description	Reset	Access
[15:0]	ADDRH	Parity Error Address High. Indicates the upper 16-bits of the address flagged as having an access parity error.	0x0	R

System IRQ Abort Enable for Interrupt 15 to Interrupt 0 Register

Address: 0x40018078, Reset: 0x0000, Name: FEE0AEN0

Table 176. Bit Descriptions for FEE0AEN0

Bits	Bit Name	Description	Reset	Access
[15:0]	FEE0AEN0	System IRQ abort enable for Interrupt 15 to Interrupt 0. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 31 to Interrupt 16 Register

Address: 0x4001807C, Reset: 0x0000, Name: FEE0AEN1

Table 177. Bit Descriptions for FEE0AEN1

Bits	Bit Name	Description	Reset	Access
[15:0]	FEE0AEN1	System IRQ abort enable for Interrupt 31 to Interrupt 16. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 47 to Interrupt 32 Register

Address: 0x40018080, Reset: 0x0000, Name: FEE0AEN2

Table 178. Bit Descriptions for FEE0AEN2

Bits	Bit Name	Description	Reset	Access
[15:0]	FEE0AEN2	System IRQ abort enable for Interrupt 47 to Interrupt 32. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 60 to Interrupt 48 Register

Address: 0x40018084, Reset: 0x0000, Name: FEE0AEN3

Table 179. Bit Descriptions for FEE0AEN3

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
[12:0]	FEE0AEN3	System IRQ abort enable for Interrupt 60 to Interrupt 48. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

Register Map for Flash Parity 1

Table 180. Flash Parity 1 Register Summary

Address	Name	Description	Reset	RW
0x40018100	FEE1STA	Status	0x0000	R
0x40018104	FEE1CON0	Command control	0x0000	RW
0x40018108	FEE1CMD	Command	0x0000	RW
0x40018110	FEE1ADR0L	Lower page address	0x0000	RW
0x40018114	FEE1ADR0H	Upper page address	0x0000	RW
0x40018118	FEE1ADR1L	Lower page address	0x0000	RW
0x4001811C	FEE1ADR1H	Upper page address	0x0000	RW
0x40018120	FEE1KEY	Key	0x0000	W
0x40018128	FEEPROL	Lower halfword of write protection	0xFFFF	RW
0x4001812C	FEEPROH	Upper halfword of write protection	0xFFFF	RW
0x40018130	FEE1SIGL	Lower halfword of signature	0x000X	R
0x40018134	FEE1SIGH	Upper halfword of signature	0x000X	R
0x40018138	FEECON1	User setup	0x000X	RW
0x40018148	FEE1ADRAL	Lower halfword of write abort address	0x000X	R
0x4001814C	FEE1ADRAH	Upper halfword of write abort address	0x000X	R
0x40018150	FEE1PARCTL	Parity control register	0x0000	RW
0x40018154	FEE1PARSTA	Parity status register	0x0000	RW1C
0x40018158	FEE1PARADRL	Parity error address low	0x0000	R
0x4001815C	FEE1PARADRH	Parity error address high	0x0000	R
0x40018178	FEE1AEN0	System IRQ abort enable for Interrupt 15 to Interrupt 0	0x0000	RW
0x4001817C	FEE1AEN1	System IRQ abort enable for Interrupt 31 to Interrupt 16	0x0000	RW
0x40018180	FEE1AEN2	System IRQ abort enable for Interrupt 47 to Interrupt 32	0x0000	RW
0x40018184	FEE1AEN3	System IRQ abort enable for Interrupt 60 to Interrupt 48	0x0000	RW

Status Register

Address: 0x40018100, Reset: 0x0000, Name: FEE1STA

Table 181. Bit Descriptions for FEE1STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
7	INIT	Initialization upload in progress. After reset, the flash controller uploads flash configuration, checks the information space signature and uploads the user write protection. This bit is asserted to 1 while this is in progress. When the upload is complete, this bit deasserts. User code does not run until this bit deasserts.	0x0	R
6	SIGNERR	Information space signature check. Information space signature check on reset error. After reset, the flash controller automatically checks the information space signature. If the signature check fails, this bit is asserted. User can check if this bit is set via serial wire only. User code does not execute if this bit is set.	0x0	R

Bits	Bit Name	Description	Reset	Access
[5:4]	CMDRES	Command result. These two bits indicate the status of a command on completion or the status of a write via the AHB. If multiple commands are executed or there are multiple writes via the AHB without a read of the status register, the first error encountered is stored. These bits clear to 00 when read. After an erase, the controller reads the corresponding word(s) to verify that the transaction completed. If data read is not all Fs, this is the resulting status. If the sign command is executed and the resulting signature does not match the data in four of the upper eight bytes of the upper page in a block then this is the resulting status. 00: successful completion of a command or a write. 01: attempted erase of a protected location. 10: read verify error. 11: indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.	0x0	RC
3	WRDONE	Write complete. This bit asserts when a write via the AHB completes. It clears when read. If there are multiple writes (or a burst write), this status bit asserts after the first long word written and stays asserted until read. If there is a burst write to flash, this bit asserts after every long word written (assuming the bit is cleared by a read after every long word written).	0x0	RC
2	CMDDONE	Command complete. This bit asserts when a command completes. It clears when read. If there are multiple commands, this status bit asserts after the first command completes and stays asserted until read.	0x0	RC
1	WRBUSY	Write busy. This bit is asserted when the flash block is executing a write via the AHB.	0x0	R
0	CMDBUSY	Command busy. This bit is asserted when the flash block is executing any command entered via the command register.	0x0	R

Command Control Register

Address: 0x40018104, Reset: 0x0000, Name: FEE1CON0

Table 182. Bit Descriptions for FEE1CON0

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved. Returns 0 when read.	0x0	R
2	WREN	Write enable. When WREN is 1, it is possible to write to flash. When it is 0, an attempt to write to a flash location results in an AHB error and the write does not take place.	0x0	RW
1	IENERR	Error interrupt enable. If this bit is set, an interrupt is generated when a command or flash write completes with an error status.	0x0	RW
0	IENCMD	Command complete interrupt enable. When set, an interrupt is generated when a command or flash write completes.	0x0	RW

Command Register

Address: 0x40018108, Reset: 0x0000, Name: FEE1CMD

The command register is used to initiate flash operations such as signature, erase, and abort. This register can be read at any time. If a command is in progress, the only write to this register is the ABORT command. All commands, including the IDLE command, result in a command complete interrupt (CMDDONE).

Table 183. Bit Descriptions for FEE1CMD

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved. Always returns 0 when read.	0x0	RW
[3:0]	CMD	IDLE: the flash controller takes no action. A command complete interrupt is issued if an idle command is initiated. It is not necessary to issue the IDLE command at the completion of any of the other command operations. ERASEPAGE: write the address of the page to be erased to the PageAddr0 register, then write this code to the FEECMD register and the flash erases the page. When the erase is complete, the flash reads every location in the page to verify all words in the page are erased. If there is a read verify error, it is indicated in the status register. To erase multiple pages, wait until a previous page erase has completed. Check the status, then issue a command to start the next page erase. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register.	0x0	RW

Bits	Bit Name	Description	Reset	Access
		FSIGN: use this command to generate a forward direction signature for a block of data. The signature is generated on a page by page basis. To generate a signature, the address of the first page (lower address) of the block is entered in the PageAddr0 register, the address of the last page (higher address) is written to the PageAddr1 register, then write this code to the command register. The signature calculation is in forward direction (that is, it starts from the LSW of lower page address (first page) and works by incrementing the address). When the command is complete, the signature is available for reading in the Sign register. The last 8 bytes of the higher address page in a block is reserved for storing the reverse and forward signature. The forward signature is placed in the last four bytes. Before entering this command, 0xF456 followed 0xF123 must be written to the key register. RSIGN: use this command to generate a reverse direction signature for a block of data. The signature is generated on a page by page basis. To generate a signature, the address of the last page (lower address) of the block is entered in the PageAddr0 register, the address of the first page (higher address) is written to the PageAddr1 register, then write this code to the command register. The signature calculation is in reverse direction (that is, it starts from MSW of higher page address (first page) and works by decrementing the address). When the command is complete, the signature is available for reading in the sign register. The last eight bytes of the higher address page in a block is reserved for storing the reverse and forward signature. The reverse signature (four bytes) is placed on four bytes that precedes forward signature (last four bytes). Before entering this command, 0xF456 followed 0xF123 must be written to the key register. For FSIGN and RSIGN commands, the lower address is always placed in the PageAddr0 register and the higher address is placed in the PageAddr1 register. If signature check is desired is on a single page, the corresponding page address is placed in both the PageAddr0 and the PageAddr1 registers. MASSERASE: erase all of user space. To enable this operation, 0xF456 followed by 0xF123 must first be written to the key register (this is to prevent accidental erases). When the mass erase is complete, the controller reads every location to verify that all locations are 0xFFFFFFFF. If there is a read verify error, it is indicated in the status register. ABORT: if this command is issued, any command currently in progress is stopped. The status indicates the command completed with an error status of CMDABORTED. Note that this is the only command that can be issued while another command is already in progress. This command can also be used to stop a write that may be in progress. If a write is aborted, the address of the location written can be read via the FEEADRA register. While the flash controller is writing one long word another long word write may be in the pipeline from the Cortex-M3 or DMA engine (depending on how the software implements writes). Therefore, both writes may need to be aborted. If a write or erase is aborted, the flash timing is violated and it is not possible to determine if the write or erase completed successfully. To enable this operation, 0xF456 followed by 0xF123 must first be written to the key register (this is to prevent accidental aborts). 0000: IDLE command. 0001: ERASEPAGE command. 0010: FSIGN command. 0011: RSIGN command. 0100: MASSERASE command. 0101: ABORT command.		

Lower Page Address Register

Address: 0x40018110, Reset: 0x0000, Name: FEE1ADR0L

Table 184. Bit Descriptions for FEE1ADR0L

Bits	Bit Name	Description	Reset	Access
[15:11]	LOW	Lower five bits of page address. Lower five bits of the address of a page in flash. Used by the erase and sign commands.	0x0	RW
[10:0]	RESERVED	Reserved. Page size is 2 kB. These 11 bits are marked as reserved so that byte addresses can be written directly to this register. The lower 11 bits of a byte address are not used because this is a page address. Returns 0x0 if read.	0x0	RW

Upper Page Address Register

Address: 0x40018114, Reset: 0x0000, Name: FEE1ADR0H

Table 185. Bit Descriptions for FEE1ADR0H

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	RW
[1:0]	HIGH	Upper bits of page address. Upper address bits of the page address.	0x0	RW

Lower Page Address Register

Address: 0x40018118, Reset: 0x0000, Name: FEE1ADR1L

Table 186. Bit Descriptions for FEE1ADR1L

Bits	Bit Name	Description	Reset	Access
[15:11]	LOW	Lower five bits of page address. Lower seven bits of the address of a page in flash. Used by the erase and sign commands.	0x0	RW
[10:0]	RESERVED	Reserved. Page size is 2 kB. These 11 bits are marked as reserved so that byte addresses can be written directly to this register. The lower 11 bits of a byte address are not used as this is a page address. Returns 0x0 if read.	0x0	RW

Upper Page Address Register

Address: 0x4001811C, Reset: 0x0000, Name: FEE1ADR1H

Table 187. Bit Descriptions for FEE1ADR1H

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0	RW
[1:0]	HIGH	Upper bits of page address. Upper address bits of the page address.	0x0	RW

Key Register

Address: 0x40018120, Reset: 0x0000, Name: FEE1KEY

Table 188. Bit Descriptions for FEE1KEY

Bits	Bit Name	Description	Reset	Access
[15:0]	KEY	Key. Enter 0xF456 followed by 0xF123. Returns 0x0 if read.	0x0	W

Lower Halfword of Write Protection Register

Address: 0x40018128, Reset: 0xFFFF, Name: FEEPROL

Table 189. Bit Descriptions for FEEPROL

Bits	Bit Name	Description	Reset	Access
[15:0]	LOW	Lower halfword of the write protection. Lower 16 bits of the write protection. Write 0 to protect a section of flash. This register is read only if the write protection in flash has been programmed.	0xFFFF	RW

Upper Halfword of Write Protection Register

Address: 0x4001812C, Reset: 0xFFFF, Name: FEEPROH

Table 190. Bit Descriptions for FEEPROH

Bits	Bit Name	Description	Reset	Access
[15:0]	HIGH	Upper halfword of write protection. Write 0 to protect a section of flash. This register is read only if the write protection in flash has been programmed.	0xFFFF	RW

Lower Halfword of Signature Register

Address: 0x40018130, Reset: 0x000X, Name: FEE1SIGL

Table 191. Bit Descriptions for FEE1SIGL

Bits	Bit Name	Description	Reset	Access
[15:0]	SIGL	Lower halfword of signature. Corresponds to SIGNATURE[15:0].	0xx	RW

Upper Halfword of Signature Register

Address: 0x40018134, Reset: 0x000X, Name: FEE1SIGH

Table 192. Bit Descriptions for FEE1SIGH

Bits	Bit Name	Description	Reset	Access
[15:0]	SIGH	Upper byte of the signature. Corresponds to SIGNATURE[31:16].	0xx	R

User Setup Register

Address: 0x40018138, Reset: 0x000X, Name: FEECON1

This register is key protected, so the key (0xF456 followed by 0xF123) must be entered in FEE1KEY. After writing to FEECON1, a 16-bit value must be written again to FEE1KEY, to re-assert the key protection.

Table 193. Bit Descriptions for FEECON1

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
0	DBG	Serial wire debug enable. If this bit is 1, access via the serial wire debug interface is enabled. If this bit is 0, access via the serial wire debug interface is disabled. The kernel sets this bit to 1 when it has finished executing, thus enabling debug access to a user.	0xx	RW

Lower Halfword of Write Abort Address Register

Address: 0x40018148, Reset: 0x000X, Name: FEE1ADRAL

Table 194. Bit Descriptions for FEE1ADRAL

Bits	Bit Name	Description	Reset	Access
[15:0]	FEEADRAL	Lower halfword of FEEADRA. If a write is aborted, this contains the address of the location written when the write was aborted. This register has relevant value only if appropriate flags in FEE1STA register are set after a write abort.	0xx	RW

Upper Halfword of Write Abort Address Register

Address: 0x4001814C, Reset: 0x000X, Name: FEE1ADRAH

Table 195. Bit Descriptions for FEE1ADRAH

Bits	Bit Name	Description	Reset	Access
[15:0]	FEEADRAH	Upper halfword of FEEADRA.	0xx	R

Parity Control Register

Address: 0x40018150, Reset: 0x0000, Name: FEE1PARCTL

Table 196. Bit Descriptions for FEE1PARCTL

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved. Returns a value of 0 when read.	0x0	R
1	PERREXEN	Parity error interrupt enable. 0: a parity error does not result in a parity interrupt. 1: a parity error results in a parity interrupt.	0x0	RW
0	PAREN	Parity enable. 0: parity checking is disabled. 1: parity checking is enabled for each ICODE and DCODE flash access.	0x0	RW

Parity Status Register

Address: 0x40018154, Reset: 0x0000, Name: FEE1PARSTA

Table 197. Bit Descriptions for FEE1PARSTA

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Returns a value of 0 when read.	0x0	R
0	PARERR	Parity error. 0: no parity error has been detected since last cleared. 1: a parity error has been detected.	0x0	RW1C

Parity Error Address Low Register

Address: 0x40018158, Reset: 0x0000, Name: FEE1PARADRL

Table 198. Bit Descriptions for FEE1PARADRL

Bits	Bit Name	Description	Reset	Access
[15:0]	ADDRL	Parity error address low. Indicates the lower 16 bits of the address flagged as having an access parity error.	0x0	R

Parity Error Address High Register

Address: 0x4001815C, Reset: 0x0000, Name: FEE1PARADRH

Table 199. Bit Descriptions for FEE1PARADRH

Bits	Bit Name	Description	Reset	Access
[15:0]	ADDRH	Parity error address high. Indicates the upper 16 bits of the address flagged as having an access parity error.	0x0	R

System IRQ Abort Enable for Interrupt 15 to Interrupt 0 Register

Address: 0x40018178, Reset: 0x0000, Name: FEE1AEN0

Table 200. Bit Descriptions for FEE1AEN0

Bits	Bit Name	Description	Reset	Access
[15:0]	FEE1AEN0	System IRQ abort enable for Interrupt 15 to Interrupt 0. To allow a system interrupt to abort a write or a command (erase, sign), write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 31 to Interrupt 16 Register

Address: 0x4001817C, Reset: 0x0000, Name: FEE1AEN1

Table 201. Bit Descriptions for FEE1AEN1

Bits	Bit Name	Description	Reset	Access
[15:0]	FEE1AEN1	System IRQ abort enable for Interrupt 31 to Interrupt 16. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 47 to Interrupt 32 Register

Address: 0x40018180, Reset: 0x0000, Name: FEE1AEN2

Table 202. Bit Descriptions for FEE1AEN2

Bits	Bit Name	Description	Reset	Access
[15:0]	FEE1AEN2	System IRQ abort enable for Interrupt 47 to Interrupt 32. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 60 to Interrupt 48 Register

Address: 0x40018184, Reset: 0x0000, Name: FEE1AEN3

Table 203. Bit Descriptions for FEE1AEN3

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
[12:0]	FEE1AEN3	System IRQ abort enable for Interrupt 60 to Interrupt 48. To allow a system interrupt to abort a write or a command (erase, sign) then write a 1 to the appropriate bit in this register.	0x0	RW

GENERAL-PURPOSE FLASH CONTROLLER

FEATURES

The ADuCM350 platform includes 16 kB of embedded flash memory for general-purpose use, such as EEPROM emulation. Unlike the 384 kB instruction flash, parity checking is not implemented on the general-purpose flash. The flash controller is connected to the bus matrix as a slave device for core and DMA access, as well as the 32-bit APB for MMR access.

The flash controller supports 16 kB of user space. Read and write to flash are executed via AHB only.

Commands supported include the following:

- Mass erase and page erase
- Generation of signatures for single or multiple pages
- Command abort
- Flash integrity
- User signature for application code
- A 128 kB failure rate of less than 0.1 ppm
- 20,000 cycle endurance with 20 ms erase, 20 µs program
- 100 year data retention at room temperature

FLASH MEMORY ORGANIZATION

The general-purpose flash memory controller supports 16 kB of user flash, as shown in Figure 61.

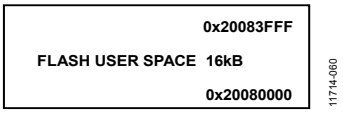


Figure 61. General-Purpose Flash Memory Map

Address Used in Registers

Figure 62. General-Purpose Flash Offset Memory Map

For the following registers, only the offset address is written or available as readback, as per Figure 62:

- GPFEEADR0L
- GPFEEADR1L
- GPFEEADRAL
- GPFEEADRAH

Full memory addresses, as in Figure 61, are not applicable for these registers. They are applicable for accesses via the AHB only.

Writing to Flash Memory

Flash is written directly by the core or bus master, using an AHB write. Only 32-bit writes are supported; 16- and 8-bit writes are not supported. Burst writes to flash are supported. Because only 0s are written to flash, masking can be used to write individual bits or bytes, if necessary. A single location can only be written twice without an erase.

If a write is followed immediately by a second write to the next location in flash, and this is in the same row as the previous write, the flash write time is faster because the controller does not need to change the row address. This is reflected in the Performance, Command Duration section.

Before performing a write to the flash memory, the GPFEESTA register must be read and checked to ensure that no other commands/writes are being executed.

FLASH INTEGRITY, SIGNATURE FEATURE

The signature is used to check the integrity of the flash device. Software can call a signature check command occasionally or whenever a new block of code is about to be executed. The signature is a 32-bit CRC with the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (IEEE 802.3).

The sign command can be used to generate a signature and check the signature of a block of code, where a block can be a single page or multiple pages. A 32-bit LFSR is used to generate the signature. The hardware assumes that the signatures for a block are stored within the upper 8 bytes of the most significant page of a block; therefore, these 8 bytes are not included when generating the signature.

The following procedure must be followed to generate a forward signature:

1. Write the start address (lower address) of the block to the PageAddr0 register.
2. Write the end address (higher address) of the block to the PageAddr1 register.
3. Write the FSIGN command to the command register.
4. When the command has completed, the signature is available in the sign register. Scan is in the forward direction; therefore, the signature is compared with the data stored in the upper 4 bytes of the uppermost page of the block.
5. If the data does not match the signature, a fail status of VerifyErr is returned in the status register.

The following procedure must be followed to generate a reverse signature:

1. Write the end address (lower address) of the block to the PageAddr0 register.
2. Write the start address (higher address) of the block to the PageAddr1 register.
3. Write the RSIGN command to the command register.
4. When the command has completed, the signature is available in the sign register. Scan is in the reverse direction; therefore, the signature is compared with the data stored in the 4 bytes that precede the upper 4 bytes of the uppermost page of the block.
5. If the data does not match the signature, a fail status of VerifyErr is returned in the status register.

While the signature is being computed, all other accesses to flash are stalled (HREADY = 0). For a 16 kB block that is 4000 reads, the signature computation for each read requires 2 cycles.

Note that PageAddr0/PageAddr1 addresses are byte addresses, but only pages must be identified; that is, the lower nine bits are ignored by the hardware, as page size of general-purpose flash is 512 bytes.

Relevant registers for running sign command in User Space 0 include the following:

- GPFEECMD: command register
- GPFEEADR0L: Page Addr0 register
- GPFEEADR1L: Page Addr1 register
- GPFEESTA: status register

BUS INTERFACE

System bus write to general-purpose flash takes 46 µs.

Command base page erase or mass erase in general-purpose flash takes approximately 21 ms.

General-purpose flash controller is in the system bus. Because erase/write operation to general-purpose flash can take a considerable amount of time (system clock cycles), it can hold the bus masters (DMA, USB, and Cortex-M3), preventing them from executing further bus operation. To avoid holding the bus masters, the following approach is taken:

- When there is a read or write access via the system bus to general-purpose flash, the bus error is returned while flash is busy executing a write operation.
- When there is a read or write access via the system bus to general-purpose flash, the bus error is returned while flash is busy executing a command operation (sign, erase, write).
- When there is a write or read access via the system bus to general-purpose flash, the bus is held for a cycle while flash is busy performing a read operation (by another master).

DMA INTERFACE FEATURE

As described in the Bus Interface section, two consecutive write requests via the system bus results in a bus error for the second write request. Due to this, DMA burst write transfers to general-purpose flash cannot be handled. In other words, writing the general-purpose flash using the DMA cannot be performed as if it were a simple memory to memory copy.

To handle burst writes to general-purpose flash via DMA, a feature is present to generate DMA requests when the flash is not busy with a command or write operation. This feature incorporates a write buffer within the general-purpose flash controller that interfaces to the DMA controller. The general-purpose flash controller requests a DMA access for data if the write buffer is empty. It stops requesting DMA data if the write buffer is full. Using this feature ensures that an optimal programming sequence is maintained.

To enable this feature, set the DMA_EN bit (Bit 3 of the GPFEECON0 register) and the appropriate DMA control bit located in GPFDMACTL (see the DMA Controller section). Then proceed to configure the DMA in peripheral request mode. Note that the USB DMA is not supported to program the general-purpose flash.

ABORT USING SYSTEM INTERRUPTS

Commands (erase, sign, and mass verify) and writes can be aborted upon receipt of an interrupt. Aborts are also possible by writing an abort command to the GPFEECMD registers. However, if flash is being programmed, and the routine controlling the programming is in flash, it is not possible to use the abort command to abort the cycle because instructions cannot be read. Therefore, the ability to abort a cycle on the assertion of any system interrupt is provided. The GPFEEAENx registers are used to enable aborts upon receipt of an interrupt.

When a command or write is aborted via a system interrupt, a CMDDONE or WRDONE status bit is asserted, and CMDRES in the status register indicates aborted.

It is not possible to abort a flash write or erase cycle without waiting for the high voltage in the flash core to discharge first; that is, ensuring a write cycle completes by waiting 6.6 µs for the high voltage to discharge. An erase cycle must finish with 6.6 µs, and a mass erase cycle must finish with a 121 µs.

Depending on the state of a write cycle when abort is asserted, the write cycle may or may not complete. If the write or erase cycle did not complete successfully, a fail status of aborted can be read in the status register.

If an immediate response to an interrupt is required during an erase or program cycle, the interrupt service routine and the interrupt vector table must be moved to SRAM for the duration of the cycle.

If the DMA engine is set up to write a block of data to flash, an interrupt can be set up to abort the current write, but the DMA engine starts the next write immediately. The interrupt causing the abort stays asserted; therefore, there are several aborted write cycles in this case before the processor can access flash.

When an abort is triggered by an interrupt, all commands are repeatedly aborted until the appropriate GPFEEAENx bit field is cleared, or the interrupt source is cleared.

POWER-DOWN INSTRUCTIONS

When powering the device down from active mode, the requirements described in this section must be met for the general-purpose flash controller.

Active Mode to Core Sleep Mode

- Nothing required because clock to flash controller is on.
- If a general-purpose flash interrupt request is asserted (see Position 55 in Table 114), the Cortex-M3 wakes up (INTSETE1 = NVIC_EN1_GPFLSH).

Active Mode to System Sleep Mode

- Check status of general-purpose flash (GPFEESTA) and enter SYS_SLEEP only if 0.

Active Mode to Hibernate Mode

If DMA is in progress,

- Disable DMA requests in the GPFEECON0 register.
- Disable DMA.
- Apply abort command.
- Wait for flash complete interrupt by enabling interrupt or poll status.
- Enter low power.

Check status of general-purpose flash (GPFEESTA). If it is not 0,

- Apply abort command.
- Wait for the command/write complete interrupt or poll status.
- Enter low power.

FLASH CRASH

For more information about flash crash, see the Flash Controller section.

PERFORMANCE, COMMAND DURATION

See the ADuCM350 data sheet. The following values are for reference purposes only:

- Direct single write access (32-bit location): 46 µs.
- Mass erase: 21 ms.
- Page erase: 21 ms.
- Direct write of a page (128 32-bit locations): 3.04 ms.
- Mass verify for all user space: 0.26 ms (16 kB and 16 MHz HCLK).
- Sign for all user space: 0.52 ms (16 kB and 16 MHz HCLK).

GENERAL-PURPOSE FLASH CONTROLLER MEMORY MAPPED REGISTERS

General-Purpose Flash Controller Register Map

Table 204. GPFLASHCTL Register Summary

Address	Name	Description	Reset	RW
0x4001C000	GPFEESTA	Status	0x0000	R
0x4001C004	GPFEECON0	Command control	0x0000	RW
0x4001C008	GPFEECMD	Command	0x0000	RW
0x4001C010	GPFEEADR0L	Lower page address	0x0000	RW
0x4001C018	GPFEEADR1L	Lower page address	0x0000	RW
0x4001C020	GPFEEKEY	Key	0x0000	W
0x4001C030	GPFEESIGL	Lower halfword of signature	0x0000	R
0x4001C034	GPFEESIGH	Upper halfword of signature	0xXXXX	R
0x4001C048	GPFEEADRAL	Lower halfword of write abort address	0xXXXX	R
0x4001C04C	GPFEEADRAH	Upper halfword of write abort address	0xXXXX	R
0x4001C078	GPFEEAEN0	System IRQ abort enable for Interrupt 15 to Interrupt 0	0x0000	RW
0x4001C07C	GPFEEAEN1	System IRQ abort enable for Interrupt 31 to Interrupt 16	0x0000	RW
0x4001C080	GPFEEAEN2	System IRQ abort enable for Interrupt 47 to Interrupt 32	0x0000	RW
0x4001C084	GPFEEAEN3	System IRQ abort enable for Interrupt 61 to Interrupt 48	0x0000	RW

Status Register

Address: 0x4001C000, Reset: 0x0000, Name: GPFEESTA

Table 205. Bit Descriptions for GPFEESTA

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[5:4]	CMDRES	Command result. These two bits indicate the status of a command on completion or the status of a write via the AHB. If multiple commands are executed or there are multiple writes via the AHB without a read of the status register, the first error encountered is stored. These bits clear to 00 when read. After an erase the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all Fs, this is the resulting status. If the sign command is executed and the resulting signature does not match the data in four of the upper eight bytes of the upper page in a block then this is the resulting status. 00: completion of a command or a write. 10: read verify error. 11: indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.	0x0	RC

Bits	Bit Name	Description	Reset	Access
3	WRDONE	Write complete. This bit asserts when a write via the AHB completes. It clears when read. If there are multiple writes (or a burst write), this status bit asserts after the first long word written and stays asserted until read. If there is a burst write to flash then this bit asserts after every long word written (assuming the bit is cleared by a read after every long word written).	0x0	RC
2	CMDDONE	Command complete. This bit asserts when a command completes. It clears when read. If there are multiple commands, this status bit asserts after the first command completes and stays asserted until read.	0x0	RC
1	WRBUSY	Write busy. This bit is asserted when the flash block is executing a write via the AHB.	0x0	R
0	CMDBUSY	Command busy. This bit is asserted when the flash block is executing any command entered via the command register.	0x0	R

Command Control Register

Address: 0x4001C004, Reset: 0x0000, Name: GPFEECON0

Table 206. Bit Descriptions for GPFEECON0

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved. Returns 0 when read.	0x0	R
3	DMA_EN	Enable DMA interface feature. A 1 indicates that the DMA interface feature for general-purpose flash is enabled. A DMA request is generated and DMA transfers can be started after setting this bit. A 0 indicates that DMA interface feature is not enabled.	0x0	RW
2	WREN	Write enable. When WREN is 1, it is possible to write to flash. When it is 0, an attempt to write to a flash location results in an AHB error and the write does not take place.	0x0	RW
1	IENERR	Error interrupt enable. If this bit is set then an interrupt is generated when a command or flash write completes with an error status.	0x0	RW
0	IENCMD	Command complete interrupt enable. When set, an interrupt is generated when a command or flash write completes.	0x0	RW

Command Register

Address: 0x4001C008, Reset: 0x0000, Name: GPFEECMD

The command register is used to initiate flash operations such as signature, erase and abort. This register can be read at any time. If a command is in progress, the only write to this register is the ABORT command. All commands, including the IDLE command, result in a command complete interrupt (CMDDONE).

Table 207. Bit Descriptions for GPFEECMD

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[3:0]	CMD	IDLE: the flash controller takes no action. A command complete interrupt is issued if an idle command is initiated. It is not necessary to issue the IDLE command at the completion of any of the other command operations. ERASEPAGE: write the address of the page to be erased to the PageAddr0 register, then write this code to the FEECMD register and the flash erases the page. When the erase has completed, the flash reads every location in the page to verify all words in the page are erased. If there is a read verify error, it is indicated in the status register. To erase multiple pages, wait until a previous page erase has completed. Check the status and then issue a command to start the next page erase. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register. FSIGN: use this command to generate a forward direction signature for a block of data. The signature is generated on a page by page basis. To generate a signature, the address of the first page (lower address) of the block is entered in the PageAddr0 register, the address of the last page (higher address) is written to the PageAddr1 register, then write this code to the command register. The signature calculation is in forward direction (that is, it starts from the LSW of the lower page address (first page) and works by incrementing the address). When the command is complete, the signature is available for reading in the sign register. The last eight bytes of the higher address page in a block is reserved for storing the reverse and forward signature. The forward signature is placed in the last four bytes. Before entering this command, 0xF456 followed 0xF123 must be written to the key register.	0x0	RW

Bits	Bit Name	Description	Reset	Access
		RSIGN: use this command to generate a reverse direction signature for a block of data. The signature is generated on a page by page basis. To generate a signature, the address of the last page (lower address) of the block is entered in the PageAddr0 register, the address of the first page (higher address) is written to the PageAddr1 register, then write this code to the command register. The signature calculation is in reverse direction (that is, it starts from the MSW of the higher page address (first page) and works by decrementing the address). When the command is complete, the signature is available for reading in the sign register. The last eight bytes of the higher address page in a block is reserved for storing the reverse and forward signature. The reverse signature (four bytes) is placed on four bytes that precedes forward signature (last four bytes). Before entering this command, 0xF456 followed 0xF123 must be written to the key register. For FSIGN and RSIGN commands, the lower address is always placed in the PageAddr0 register and the higher address is placed in the PageAddr1 register. If signature check is on a single page, the corresponding page address is placed in both the PageAddr0 and PageAddr1 registers. MASSERASE: erase all of user space. To enable this operation, 0xF456 followed by 0xF123 must first be written to the key register (this is to prevent accidental erases). When the mass erase has completed, the controller reads every location to verify that all locations are 0xFFFFFFFF. If there is a read verify error, it is indicated in the status register. ABORT: If this command is issued, any command currently in progress is stopped. The status indicates command completed with an error status of CMDABORTED. Note that this is the only command that can be issued while another command is already in progress. This command can also be used to stop a write that may be in progress. If a write is aborted, the address of the location written can be read via the FEEADRA register. While the flash controller is writing one long word, another long word write may be in the pipeline from the Cortex-M3 or DMA engine (depending on how the software implements writes). Therefore, both writes may need to be aborted. If a write or erase is aborted, the flash timing is violated and it is not possible to determine if the write or erase completed. To enable this operation, 0xF456 followed by 0xF123 must first be written to the key register (this is to prevent accidental aborts). 0000: IDLE command. 0001: ERASEPAGE command. 0010: FSIGN command. 0011: RSIGN command. 0100: MASSERASE command. 0101: ABORT command.		

Lower Page Address Register

Address: 0x4001C010, Reset: 0x0000, Name: GPFEEADR0L

Table 208. Bit Descriptions for GPFEEADR0L

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved. Returns 0 if read.	0x0	R
[13:9]	LOW	Lower five bits of page address. Lower five bits of the address of a page in flash. Used by the erase and sign commands.	0x0	RW
[8:0]	RESERVED	Reserved. These nine bits are marked as reserved so that byte addresses can be written directly to this register. The lower nine bits of a byte address are not used as this is a page address. Returns 0x0 if read.	0x0	R

Lower Page Address Register

Address: 0x4001C018, Reset: 0x0000, Name: GPFEEADR1L

Table 209. Bit Descriptions for GPFEEADR1L

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R
[13:9]	LOW	Lower five bits of page address. Lower five bits of the address of a page in flash. Used by the erase and sign commands.	0x0	RW
[8:0]	RESERVED	Reserved. These nine bits are marked as reserved so that byte addresses can be written directly to this register. The lower nine bits of a byte address are not used as this is a page address. Returns 0x0 if read.	0x0	R

Key Register

Address: 0x4001C020, Reset: 0x0000, Name: GPFEEKEY

Table 210. Bit Descriptions for GPFEEKEY

Bits	Bit Name	Description	Reset	Access
[15:0]	KEY	Key. Enter 0xF456 followed by 0xF123. Returns 0x0 if read.	0x0	W

Lower Halfword of Signature Register

Address: 0x4001C030, Reset: 0x0000, Name: GPFEESIGL

Table 211. Bit Descriptions for GPFEESIGL

Bits	Bit Name	Description	Reset	Access
[15:0]	SIGL	Lower halfword of signature. Corresponds to SIGNATURE[15:0].	0x0	R

Upper Halfword of Signature Register

Address: 0x4001C034, Reset: 0x000X, Name: GPFEESIGH

Table 212. Bit Descriptions for GPFEESIGH

Bits	Bit Name	Description	Reset	Access
[15:0]	SIGH	Upper byte of the signature. Corresponds to SIGNATURE[31:16].	0xx	R

Lower Halfword of Write Abort Address Register

Address: 0x4001C048, Reset: 0x000X, Name: GPFEEADRAL

Table 213. Bit Descriptions for GPFEEADRAL

Bits	Bit Name	Description	Reset	Access
[15:0]	FEEADRAL	Lower halfword of FEEADRA. If a write is aborted, this contains the address of the location been written when the write was aborted.	0xx	RW

Upper Halfword of Write Abort Address Register

Address: 0x4001C04C, Reset: 0x000X, Name: GPFEEADRAH

Table 214. Bit Descriptions for GPFEEADRAH

Bits	Bit Name	Description	Reset	Access
[15:0]	FEEADRAH	Upper halfword of FEEADRA.	0xx	R

System IRQ Abort Enable for Interrupt 15 to Interrupt 0 Register

Address: 0x4001C078, Reset: 0x0000, Name: GPFEEAEN0

Table 215. Bit Descriptions for GPFEEAEN0

Bits	Bit Name	Description	Reset	Access
[15:0]	GPFEEAEN0	System IRQ abort enable for Interrupt 15 to Interrupt 0. To allow a system interrupt to abort a write or a command (erase, sign), write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 31 to Interrupt 16 Register

Address: 0x4001C07C, Reset: 0x0000, Name: GPFEEAEN1

Table 216. Bit Descriptions for GPFEEAEN1

Bits	Bit Name	Description	Reset	Access
[15:0]	GPFEEAEN1	System IRQ abort enable for Interrupt 31 to Interrupt 16. To allow a system interrupt to abort a write or a command (erase, sign), write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 47 to Interrupt 32 Register

Address: 0x4001C080, Reset: 0x0000, Name: GPFEEAEN2

Table 217. Bit Descriptions for GPFEEAEN2

Bits	Bit Name	Description	Reset	Access
[15:0]	GPFEEAEN2	System IRQ abort enable for Interrupt 47 to Interrupt 32. To allow a system interrupt to abort a write or a command (erase, sign), write a 1 to the appropriate bit in this register.	0x0	RW

System IRQ Abort Enable for Interrupt 60 to Interrupt 48 Register

Address: 0x4001C084, Reset: 0x0000, Name: GPFEEAEN3

Table 218. Bit Descriptions for GPFEEAEN3

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
[12:0]	GPFEEAEN3	System IRQ abort enable for Interrupt 60 to Interrupt 48. To allow a system interrupt to abort a write or a command (erase, sign), write a 1 to the appropriate bit in this register.	0x0	RW

CRC ACCELERATOR

INTRODUCTION

The CRC accelerator can be used to compute the CRC for a block of memory locations. The exact memory location can be in the SRAM, flash, or any combination of memory mapped registers. The CRC accelerator generates a checksum that can be used to compare it with an expected signature. The final CRC comparison is performed by the processor.

FEATURES

The main features of the CRC include the following:

- Generates a CRC signature for a block of data.
- Operates on 32-bit data.
- Performs data mirroring.
- Uses an IEEE CRC 32-bit polynomial.
- Includes integrated DMA support.

Control for address decrement/increment options for computing the CRC on a block of memory is via the DMA controller. For details about these options, refer to the Address Decrement subsection within the DMA Controller section. Figure 63 shows the major components of the CRC accelerator.

BLOCK DIAGRAM

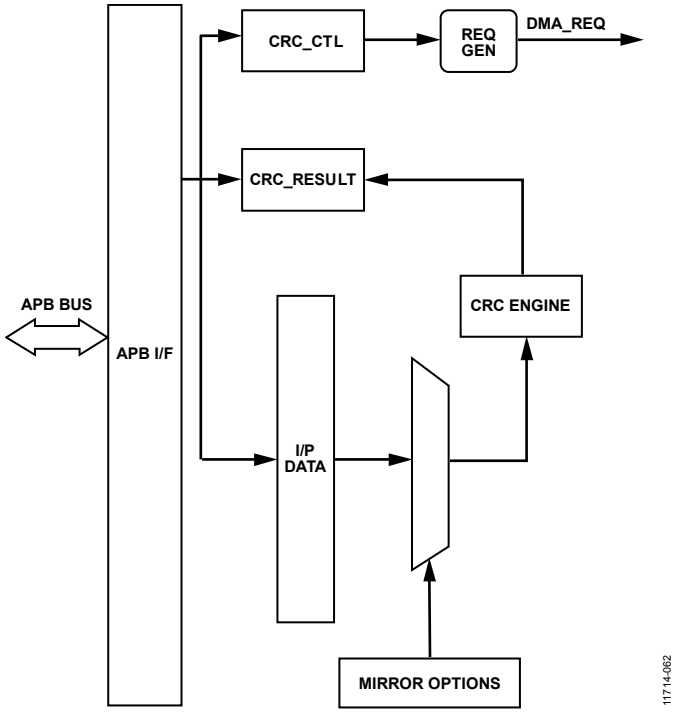


Figure 63. CRC Accelerator Block Diagram

OPERATION

The accelerator calculates CRC on the data stream it receives, which is read into the block using the central DMA engine or by the processor directly. The ADuCM350 dedicates a DMA channel for this operation. A DMA driven data transfer is recommended when the CRC is to be computed for contiguous memory or register locations. If the CRC must be computed for noncontiguous memory or register locations, a processor driven transfer is recommended.

The accelerator takes two clock cycles to compute CRC for one data item. The processor can send one data item for every two clock cycles via the 32-bit APB; therefore, the Cortex-M3 does not need to wait between data transfers to the CRC accelerator.

CRC works on 32-bit data-words only. For data-words less than 32 bits in size, the processor packs the data into 32-bit data units. Data mirroring on the input data can be done at bit, byte, and word level before the CRC engine uses it by setting the BITMIRR, BYTMIRR, and W16SWP bits, respectively, in CRC_CTL register.

The CRC engine uses the following CRC polynomial (IEEE 802.3):

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

When the CRC engine is operating, a 32-bit CRC algorithm is applied to the incoming data stream being written to the CRC_IPDATA register using DMA or processor. For every new word of data received, the CRC is computed and the CRC_RESULT register is updated. The CRC engine uses the current CRC_RESULT value for generating the next CRC_RESULT value when a new data-word is received. To have a clean start for the next block of data, the CRC_RESULT register can either be cleared to 0x0000_0000 or set to 0xFFFF_FFFF. This can be done using the AUTORST option and STRT_CRC in the CRC_CTL register.

When a message or data is received from an external source, it is recommended that the expected signature be computed using an external agent. In the event that the message or data is to be transmitted, the user can use the CRC engine to generate the expected signature and transmit this with the message to be used by the external agent to validate the transmitted message or data.

PROGRAMMING MODEL

This block is provided to calculate the CRC signature over a block of data in the background while the core is performing other tasks. The CRC block supports two modes of CRC calculation: core access and DMA access. The programming details of these two modes follow.

CORE ACCESS

1. Kick off the CRC accelerator block by writing into the CRC_CTL register. (The following steps, Step a to Step e, require only a single write to the CRC_CTL register.)
   a. Set the BLKEN bit high. This bit has priority over other bits in the CRC_CTL.
   b. The W16SWP, BYMIRR, and BITMIRR bits provide the application with different mirror options. The details of these mirror options can be found in Table 219.
   c. The AUTORST bits set the initial seed for the CRC calculation. To have a clean start, use the 0 or 1 option.
   d. Because the intent of the core access is to provide the CRC data block using the core, clear the DMAREQEN bit.
   e. Set STRT_CRC to 1 to indicate the start of a new data block. The AUTORST bits are effective only when the STRT_CRC bit is set.
2. Pump in the data to the CRC accelerator block. The core can now start sending data to the CRC block by writing into the CRC_IPDATA register. The CRC accelerator keeps calculating the CRC of the data being written to the CRC_IPDATA register. The CRC block operates fast; therefore, the core need not wait in between data transfers to the CRC accelerator.
3. Monitor for the completion of the CRC. It is the responsibility of the application function(s) to count the number of words written to the CRC block. After all the words are written, the application can read the CRC_RESULT register.
4. Read the CRC_RESULT register and store it in a desired location.
5. Calculate the CRC on the next data block. To calculate the CRC on the next block of data, repeat Step a to Step d.
6. Disable the CRC accelerator block by clearing the BLKEN bit.

DMA ACCESS

The CRC accelerator block supports hardware DMA as well as software DMA. Use of hardware DMA is recommended when other peripherals, such as SPI and I^2C, which are on the higher priority channels, are being used. Software DMA is recommended when higher priority channels are not used. For example, when the system is intended to be shut down, the CRC accelerator can use software DMA with higher R_POWER to facilitate the CRC calculation at high speeds.

1. Set up the DMA channel. The CRC accelerator block a has dedicated hardware DMA channel. Set up the DMA channel as required. The DST_END_PNTR value is the CRC_IPDATA register address. Data size is word. Use the destination no increment option for the channel. Use either software or hardware DMA based on the application. See the DMA Controller section for details about programming the DMA.
2. Enable the accelerator function by writing to the CRC_CTL register. (The following steps, Step a to Step e, require only a single write to the CRC_CTL register.)
   a. Set the BLKEN bit high. This bit has priority over other bits in the CRC_CTL register.
   b. The W16SWP, BYMIRR, and BITMIRR bits provide the application with different mirror options. The details of these mirror options can be found in Table 219.
   c. The AUTORST bits set the initial seed for the CRC calculation. To have a clean start, use the 0 or 1 option.
   d. Because the intent of the DMA access is to provide the data block using DMA, set the DMAREQEN bit accordingly. DMAREQEN must be high when using hardware DMA and low when using software DMA.
   e. Set the STRT_CRC to 1 to indicate the start of a new data block. The AUTORST bits are effective only when the STRT_CRC is set.
3. Pump in the CRC data to the accelerator block. The DMA can now start sending the CRC data by writing to the CRC_IPDATA. The CRC accelerator block keeps calculating the CRC of the data being written to the CRC_IPDATA.

4. Monitor for the completion of the CRC. A DMA_DONE interrupt signal of the DMA channel indicates the completion of the CRC block. Disable the CRC block by clearing the BLKEN bit to be safe and to ensure that additional requests are not generated by the CRC block to the DMA block.
5. Repeat Step a to Step d until all the data is sent to the accelerator block.
6. Read the CRC_RESULT register and store it in the desired location.
7. Calculate CRC on the next data block. To calculate the CRC on the next block of data, repeat Step a to Step e.
8. Disable the CRC accelerator block by clearing the BLKEN bit.

Note that for DMA accesses, it is recommended to disable the CRC block by clearing the BLKEN bit within the DMA ISR routine.

RECOMMENDATIONS

When using the CRC accelerator, adhere to the following recommendations and considerations:

- Disable the CRC accelerator block by clearing the BLKEN bit when the block is not being used.
- All the transactions to the CRC accelerator block when the BLKEN bit is cleared are ignored.
- Note that the AUTORST option is effective only when the STRT_CRC is high.
- Note that the STRT_CRC bit automatically clears itself in one clock cycle. When read back, this bit is always low.
- Use the same mirror options for the entire block of data. If the application changes the mirror options in the middle of a data block, it must know precisely the mirror option used for each data set to reconstruct the correct signature.
- CRC_RESULT is a read/write register on the APB side. When CRC is being calculated, the CRC_RESULT register must not be written. Overwriting CRC_RESULT makes the CRC signature unusable.
- The procedure to calculate the CRC signature with a user defined seed is as follows:

    1. Enable CRC accelerator block. Write the CRC_CTL with BLKEN bit high.
    2. Set up the CRC signature seed. Write the CRC_RESULT register with the desired seed value.
    3. Calculate the CRC signature using DMA or core with AUTORST Option 3.

- The CRC accelerator is considerably fast using higher R_POWER values only in software DMA as compared to hardware DMA, where only R_POWER = 0 is allowed. Using higher R_POWER in software DMA may result in poor performance over other channels; therefore, care must be taken while using higher R_POWER values.
- Software DMA is much faster than hardware DMA for the CRC accelerator block. With software DMA, a suitable R_POWER value can be selected, and the corresponding DMA channel for the CRC accelerator can be put in auto request/memory scatter gather mode. This allows faster CRC calculations and does not significantly affect other low speed channels.
- While using software DMA, the DMAREQEN bit must be low and requests must be generated only via the software for the channel used by the CRC accelerator. If both the hardware and software DMA operate simultaneously, the behavior of the DMA channel for the CRC accelerator is indeterminate.
- The CRC accelerator block supports automatic request and memory scatter gather mode of DMA only while using software DMA. This feature enables the user to program the CRC calculation for a large chunk of data in a single action while the CRC calculation continues in the background requiring no intervention. The final result can be read after the DMA channel done interrupt.
- For DMA accesses, the application must wait for the channel done interrupt for the CRC channel (13 in case of hardware DMA and the specific channel assigned in case of software DMA) before changing the control word in the CRC_CTL register. If the control word is changed in between an ongoing data block transaction from DMA, the result of the previous transaction must be discarded.
- The CRC accelerator block is not retained in hibernate mode. If the application needs to use the CRC value after coming out of hibernate mode, store the CRC_RESULT in a retained region such as SRAM.

MIRRORING OPTIONS

The W16SWP, BYTMIRR, and BITMIRR bits of the CRC_CTL register dictate the sequence of the bits on which the CRC is calculated. Table 219 lists the details of all the mirroring options used within this block. Assume DIN[31:0] shown in the Table 219 is the data being written to the CRC_IPDATA register and CIN[31:0] is the data after the mirroring of the data. The serial engine calculates CIN[31:0], starting with the MSB bit and ending with LSB bit in sequence, that is, CIN[31], CIN[30], …, CIN[1], CIN[0] in order.

Table 219. Mirroring Options

W16SWP	BYTMIRR	BITMIRR	Input Data, DIN[31:0]	CRC Input Data, CIN[31:0]
0	0	0	DIN[31:0]	CIN[31:0] = DIN[31:0]
0	0	1	DIN[31:0]	CIN[31:0] = DIN[24:31], DIN[16:23], DIN[8:15], DIN[0:7]
0	1	0	DIN[31:0]	CIN[31:0] = DIN[23:16], DIN[31:24], DIN[7:0], DIN[15:8]
0	1	1	DIN[31:0]	CIN[31:0] = DIN[16:23], DIN[24:31], DIN[0:7], DIN[8:15]
1	0	0	DIN[31:0]	CIN[31:0] = DIN[15:0], DIN[31:16]
1	0	1	DIN[31:0]	CIN[31:0] = DIN[8:15], DIN[0:7], DIN[24:31], DIN[16:23]
1	1	0	DIN[31:0]	CIN[31:0] = DIN[7:0], DIN[15:8], DIN[23:16], DIN[31:24]
1	1	1	DIN[31:0]	CIN[31:0] = DIN[0:7], DIN[8:15], DIN[16:23], DIN[24:31]

SIGNATURE PROCESSING

The ADuCM350 flash controller signature generation processes data in aligned 32-bit words, whereas some published CRC algorithms process one byte at a time. This means that comparison between the two is dependent on the endianness of the device. On a big endian device the two are equivalent, but on an little endian device, such as this, the algorithm yields a different result due to the bytes within each word being processed in a different order. The flash controller effectively hardcodes the W16SWP, BYTMIRR, and BITMIRR bits to 1, which results in a complete bit reversal of the input data as compared to a PC-based CRC tool (big endian). This accounts for the little endian implementation of the 32-bit machine of the device.

This implementation of the ADuCM350 does not impair the robustness of the flash signature checking because the integrity of the CRC is independent of the order in which bytes are processed.

The ADuCM350 CRC generator block also processes data in aligned 32-bit words, but it includes byte reversal flags, which allow it to be configured appropriately to the endianness expected by a specific algorithm. The flash controller does not include such configurability This is a reasonable decision because the CRC generator may be used to verify data which has originated elsewhere, or from a standard protocol, whereas the results of the flash controller are only compared against signature values that have been explicitly prepared for that purpose.

As an example, using 0xFEEDBEEF to match the Lammert Bies online CRC calculator,

1. Set BITMIRR = 1.
2. Set STRT_CRC (to clear the result to the initial seed of 0xFFFFFFFF).
3. Take the result of 0xDA7ED3CA and reverse the bit order, which results in 0x53CB7E5B.
4. Take the ones complement, which results in 0xAC3481A4.

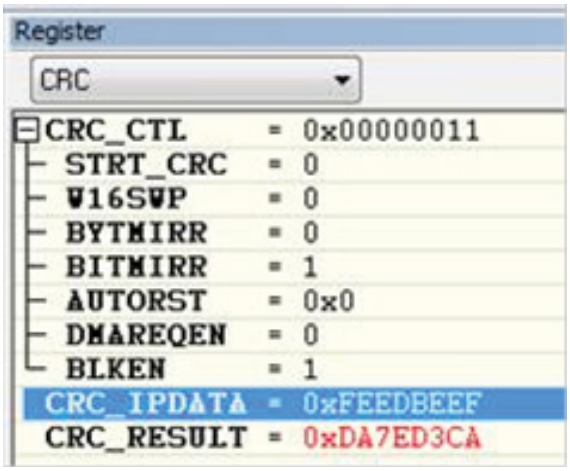


Figure 64. CRC Signal Processing Example

CRC ACCELERATOR MEMORY MAPPED REGISTERS

CRC Accelerator Register Map

Table 220. CRC Accelerator Register Summary

Address	Name	Description	Reset	RW
0x4002C000	CRC_CTL	CRC control register	0x00000000	RW
0x4002C004	CRC_IPDATA	Input data register	0x00000000	RW
0x4002C008	CRC_RESULT	CRC result register	0x00000000	RW

CRC Control Register

Address: 0x4002C000, Reset: 0x00000000, Name: CRC_CTL

Table 221. Bit Descriptions for CRC_CTL

Bits	Bit Name	Description	Reset	Access
[31:8]	RESERVED	Reserved.	0x000000	RESERVED
7	STRT_CRC	Start of CRC calculation for new block of data. 0: no effect on the CRC calculation going on. 1: initializes the CRC_RESULT register value given by AUTORST bits.	0x0	RWAC
6	W16SWP	Word16 swap. 0: Word16 swap disabled. 1: Word16 swap enabled.	0x0	RW
5	BYTMIRR	Byte mirroring. 0: byte mirroring is disabled. 1: byte mirroring is enabled.	0x0	RW
4	BITMIRR	Bit mirroring. 0: bit mirroring is disabled. 1: bit mirroring is enabled.	0x0	RW
[3:2]	AUTORST	Automatically reset the current result register. 00: automatically reset the CRC_RESULT register to 0xFFFF_FFFF. 01: automatically reset the CRC_RESULT register to 0x0000_0000. 10: no automatic reset of CRC_RESULT (default). 11: not available.	0x0	RW
1	DMAREQEN	Hardware DMA request enable. 0: DMA request is not set for new data item. 1: DMA request is set for new data item.	0x0	RW
0	BLKEN	CRC peripheral enable. 0: CRC peripheral is disabled. 1: CRC peripheral is enabled.	0x0	RW

Input Data Register

Address: 0x4002C004, Reset: 0x00000000, Name: CRC_IPDATA

Table 222. Bit Descriptions for CRC_IPDATA

Bits	Bit Name	Description	Reset	Access
[31:0]	CRC_IPDATA	Input data register. Holds 32-bit data that can be written by processor or DMA controller.	0x0	RW

CRC Result Register

Address: 0x4002C008, Reset: 0x00000000, Name: CRC_RESULT

Table 223. Bit Descriptions for CRC_RESULT

Bits	Bit Name	Description	Reset	Access
[31:0]	CRC_RESULT	CRC result.	0x0	RW

RANDOM NUMBER GENERATOR

DESCRIPTION

The random number generator (RNG) is used during operations where nondeterministic values are required. This may include generating challenges for secure communications. The generator can be run multiple times to generate a sufficient number of bits for the strength of the intended operation.

RNG ARCHITECTURE

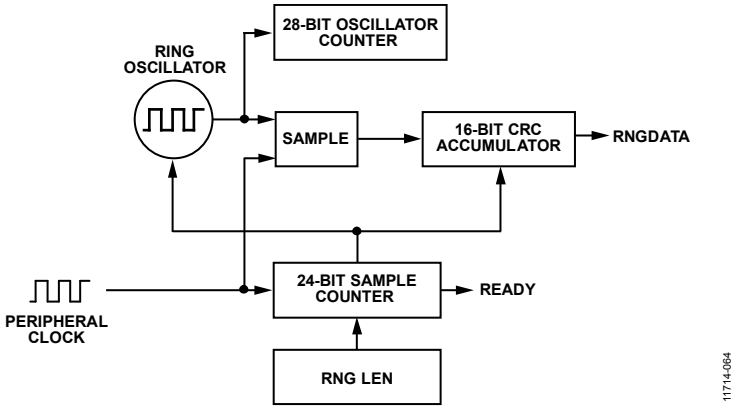


Figure 65. Structure of the RNG

The RNG is based on a sampled asynchronous clock (in this implementation a ring oscillator). A CRC is used to accumulate a programmable number of samples to gather a sufficient amount of entropy.

The RNG is enabled via its control register. The length register specifies the number of samples for which the RNG must run to accumulate sufficient entropy. One sample is obtained on each peripheral clock. After the RNG has accumulated the programmed number of samples, the 16-bit CRC value can be read. Software can poll a status bit or be interrupted when the RNG is ready.

Any number of iterations can be run. As an example, 112 bits of entropy can be obtained by running the RNG seven times.

RNG OSCILLATOR COUNTER

The RNG peripheral includes an oscillator counter. This counter is disabled by default to conserve power and is not needed to generate random numbers. This on-chip counter can be used to characterize the sampling jitter that is difficult to measure off chip. The sample jitter is the source of entropy for the random number generator.

By counting the number of ring oscillator clocks over a given sampling period, the frequency ratio of the ring oscillator to the peripheral sampling clock can be determined (and thus the ring oscillator frequency can be determined if the peripheral clock frequency is known).

$$f_{OSCCLK} = f_{PCLK} \times \frac{OSCCNT}{SAMPCNT}$$

where:
f_{OSCCLK} is the ring oscillator frequency.
f_{PCLK} is the PCLK frequency.
OSCCNT is the number of oscillator clocks.
SMPCNT is the length of the sampling time.

The length of sampling time (SAMPCNT) used by the sample counter can be determined from the RNGLEN register.

The sample jitter can be determined by calculating the standard deviation of multiple oscillator count values. Pseudocode for an efficient loop to determine the average oscillator count and jitter follows:

```
sum=0; sum_sqr=0;
for(i=0;i<N;i++) {
    gen_rng();
    sum+=osc_cnt; sum_sqr+=osc_cnt*osc_cnt;
}
avg=sum/N;
std=sqrt((sum_sqr-avg*sum)/(N-1));
```

Note that the code has been simplified for clarity. If implemented in C, the following points must be considered:

- The sum and sum_sqr variables must be integer data types. Floating point variables may lose precision if the sums accumulate for a sufficient length to overflow the fractional portion of the variable truncating the result. It is desirable to keep the entire fractional portion, so the exponent of a real data type can be excluded.
- The sum variable requires log2(osc_cnt) + log2(N) bits of storage.
- The sum_sqr variable requires 2 × log2(osc_cnt) + log2(N) bits of storage.
- The squaring operation (osc_cnt × osc_cnt and avg × avg) likely needs a cast to a type with size at least 2 × log2(osc_cnt).
- The avg and std variables can take on fractional values and must be real or fixed point data types.
- The division needs a cast to a fractional data type.

For accurate jitter measurements, the standard deviation of OSCNT must be at least 1. If it is less than 1, then SAMPCNT can be increased.

The jitter of the oscillator counter encapsulates both the jitter of the ring oscillator and the jitter of the peripheral clock. This can be used to determine sample jitter. If the jitter of the peripheral clock is known, the jitter of the oscillator clock can be determined using the following equation:

$$\frac{\sigma_{OSCCNT}}{\sqrt{SAMPCNT}} \times \frac{SAMPCNT}{OSCCNT} \times \frac{1}{f_{PCLK}} = \sigma_{SAMPLE} = \sqrt{\sigma_{PCLK}^2 + \sigma_{OSCCLK}^2 \times \frac{OSCCNT}{SAMPCNT}}$$

where:
σ_{OSCCNT} is the jitter of the oscillator counter.
σ_{PCLK} is the jitter of PCLK.
σ_{OSCCLK} is the jitter of the oscillator clock.

RNG ENTROPY AND SUPRISAL

The jitter of the sampling clocks is the source of entropy for the random number generator. Noise in the transistors of the oscillator contributes to jitter. Accumulated long-term jitter increases with the square root of time or the number of clocks.

$$\sigma_{TOTAL} = \sigma_{CLK}\sqrt{N}$$

where:
σ_{TOTAL} is the total jitter.
σ_{CLK} is the clock jitter.
N is the number of clocks.

The number of samples that must be accumulated to achieve ideal entropy of 1.0/bit depends on the amount of jitter in the system. Entropy and suprisal are calculated as follows:

$$Entropy = -\sum p_i \log_2(p_i)$$

$$Minimum\ Suprisal = \min(-\log_2(p_i))$$

The probability of seeing each number must be the same or uniform for all numbers for an ideal generator. A deficient generator outputs some numbers more frequently than others. This random number generator was designed to retain no state such that entropy can be measured and quantified in this manner. For each random number generated, the CRC is reset, and the ring oscillator starts up in the same phase. If there is insufficient entropy in the system, the generator outputs the same number more frequently. This is by design and makes entropy assessment possible. Insufficient entropy can be observed by setting SAMPCNT to a low value (some random numbers appear more frequently than others). SAMPCNT must be increased until the probability of seeing each number is uniform.

The entropy or minimum suprisal of the RNG register can be computed by tallying a histogram of generated random numbers and calculating the probability of each number occurring. If there is insufficient memory in the system to accumulate the entire histogram table, the histogram can be computed in sections (the histogram statistics must be consistent over multiple passes). The entropy measure is the average amount of randomness and is often used in the analysis of compression algorithms or communication channels. The minimum suprisal is a measure of the minimum amount of randomness, which is more conservative than entropy and must be the preferable measure for quantifying randomness that is used in security applications.

The minimum value for SAMPCNT (set by RNGLEN) that is needed to obtain the ideal minimum suprisal for the 16-bit accumulator can be determined using the following equation. A conservative design sets SAMPCNT at least as large as this, if not greater (to account for variations in jitter across various operating conditions).

$$SAMPCNT_{MIN} = \frac{1}{\sigma^2_{SAMPLE} \times f_{OSC}{}^2}$$

If an attacker can physically tamper the system and has control of the peripheral clock, the jitter due to the peripheral clock must be removed from the sample jitter in the previous calculation. This assumes an attacker can replace the crystal with a low jitter clock source. A physical attacker may also use better voltage supplies that can minimize the amount of noise in the system. If an attacker can remove or minimize the peripheral clock jitter, the entropy source of the random number generator is reduced. The system must rely solely on the jitter of the ring oscillator and conservatively assume the jitter of the peripheral clock to be 0.

TIMER MODE

The sample counter in the RNG effectively acts as a one-shot timer. If the user does not need the random number generator during a portion of an application, the RNG can be used as a simple one-shot timer. The count value cannot be read, but an interrupt can be generated, and a status bit can be set.

When used as a timer, the ring oscillator and entropy accumulator (CRC) running consume power. The TMRMODE bit in the RNGCTL register can be used to disable the ring oscillator and the entropy accumulator when using the RNG as a simple one-shot timer.

POWER-DOWN CONSIDERATIONS

When entering hibernate mode, disable the RNGEN bit in the RNGCTL register.

RNG MEMORY MAPPED REGISTERS

RNG Register Map

Table 224. RNG Register Summary

Address	Name	Description	Reset	RW
0x40006000	RNGCTL	RNG control register	0x0000	RW
0x40006004	RNGLEN	RNG sample length register	0x6A00	RW
0x40006008	RNGSTAT	RNG status register	0x0000	RW
0x4000600C	RNGDATA	RNG data register	0xFFFF	R
0x40006010	RNGCNTL	RNG oscillator count low	0x0000	R
0x40006014	RNGCNTH	RNG oscillator count high	0x0000	R

RNG Control Register

Address: 0x40006000, Reset: 0x0000, Name: RNGCTL

The RNGCTL register is used to enable the random number generator.

Table 225. Bit Descriptions for RNGCTL

Bits	Bit Name	Description	Reset	Access
2	TMRMODE	Timer mode. This control bit disables the ring oscillator and CRC entropy accumulator when the random number generator is enabled. This can be used to minimize power consumption when using the RNG as a simple timer. 0: RNG mode. 1: timer mode.	0x0	RW
1	CNTEN	Oscillator counter enable. This bit enables the oscillator counter when generating a random number. 0: oscillator counter disabled. 1: oscillator counter enabled.	0x0	RW
0	RNGEN	RNG enable. When RNGEN is set and RNGRDY is clear, the ring oscillator is powered up and the number of samples defined by RNGLEN accumulates in the RNGDATA register. 0: disable the RNG. 1: enable the RNG.	0x0	RW

RNG Sample Length Register

Address: 0x40006004, Reset: 0x6A00, Name: RNGLEN

The RNGLEN register defines the number of samples to accumulate in the CRC register when generating a random number. The number of samples accumulated is the counter reload value scaled by $2^{PRESCALER}$.

Table 226. Bit Descriptions for RNGLEN

Bits	Bit Name	Description	Reset	Access
[15:12]	LENPRE	Prescaler for the sample counter. The sample counter reload value LENRLD is scaled by 2^{LENPRE}. The prescaler is a 12-bit counter. Valid values for the prescaler are 0 to 12. Values greater than 12 saturate at the maximum prescaler value.	0x6	RW
[11:0]	LENRLD	Reload value for the sample counter. Defines the number of samples to accumulate in the CRC when generating a random number.	0xA00	RW

RNG Status Register

Address: 0x40006008, Reset: 0x0000, Name: RNGSTAT

The RNGSTAT register indicates when the RNG has finished generating a random number.

Table 227. Bit Descriptions for RNGSTAT

Bits	Bit Name	Description	Reset	Access
0	RNGRDY	Random number ready. This bit indicates when the sample counter has expired. An interrupt is generated when this bit is set. The ring oscillator is stopped when this bit is set to conserve power. This bit is automatically cleared when the RNGDATA register is read and the CPU is not stopped in debug halt. This bit can also be written with one to clear. 0: RNGDATA not ready. 1: RNGDATA register is ready to be read.	0x0	RW1C

RNG Data Register

Address: 0x4000600C, Reset: 0xFFFF, Name: RNGDATA

RNGDATA register provides the CPU with read only access of the entropy accumulator (16-bit CRC).

Table 228. Bit Descriptions for RNGDATA

Bits	Bit Name	Description	Reset	Access
[15:0]	RNGDATA	Value of the CRC accumulator. The contents of this register are valid when the RNGRDY bit is set. This register is reset when the RNGRDY bit is cleared. The RNGRDY bit is automatically cleared when this register is read and the CPU is not in debug halt. Reading this register by the CPU when RNGEN is set causes a new random number to be generated.	0xFFFF	R

Oscillator Count Low Register

Address: 0x40006010, Reset: 0x0000, Name: RNGCNTL

When enabled, the oscillator counter counts the number of ring oscillator cycles that occur during the generation of a random number. The oscillator counter is 28 bits. The oscillator counter saturates at the maximum value to prevent overflow.

Table 229. Bit Descriptions for RNGCNTL

Bits	Bit Name	Description	Reset	Access
[15:0]	RNGCNTL	Lower bits of oscillator count. This register is only valid when RNGRDY is set and CNTEN was set when the random number was generated.	0x0	R

Oscillator Count High Register

Address: 0x40006014, Reset: 0x0000, Name: RNGCNTH

Table 230. Bit Descriptions for RNGCNTH

Bits	Bit Name	Description	Reset	Access
[11:0]	RNGCNTH	Upper bits of oscillator count.	0x0	R

UNIVERSAL SERIAL BUS CONTROLLER

FEATURES

The universal serial bus (USB) is an industry standard serial bus used to manage communication between devices and a host controller attached to a computing device. The USB port on the ADuCM350 is a full speed (12 Mbps) device.

The USB controller is connected to the system via the advanced high performance bus (AHB). It has a built in DMA master to transfer data between the USB port and system memory.

The USB features available on the ADuCM350 include the following:

- USB 2.0 full speed compliant
- Supports bulk, isochronous, interrupt, and control modes
- Seven hardware endpoints
  - One control endpoint (Endpoint 0)
  - Three Tx data endpoints (Tx Endpoint 1, Tx Endpoint 2, and Tx Endpoint 3)
  - Three Rx data endpoints (Rx Endpoint 1, Rx Endpoint 2, and Rx Endpoint 3)
- Available maximum FIFO size for each endpoint (in both Rx and Tx directions):
  - Endpoint 0: 64 bytes
  - Endpoint 1: 128 bytes
  - Endpoint 2: 256 bytes
  - Endpoint 3: 512 bytes
- Integrated 2-channel DMA
- Suspend and wake-up support
- Integrated full speed PHY

The ADuCM350 USB controller hardware is supplemented by a complete set of USB device class drivers to provide complete USB functionality using Micrium USB 4.0 stack.

The supported classes are as follows:

- Personal healthcare device class (PHDC)
- Mass storage class (MSC)
- Communications device class/abstract class module (CDC/ACM)
- Human interface class (HID)

ARCHITECTURE

Figure 66 shows a top level diagram of the USB subsystem. The module consists of the USB controller, USB PHY, USB RAM, and a 2-channel DMA. The USB PHY includes the analog transceiver that drives and monitors the USB bus. The USB controller encodes and decodes USB packets and uses the AHB interfaces to move data between the controller and system memory. Each endpoint contains a dedicated FIFO (implemented as SRAM) to buffer packets. The DMA can be used to move data directly between the USB and system memory.

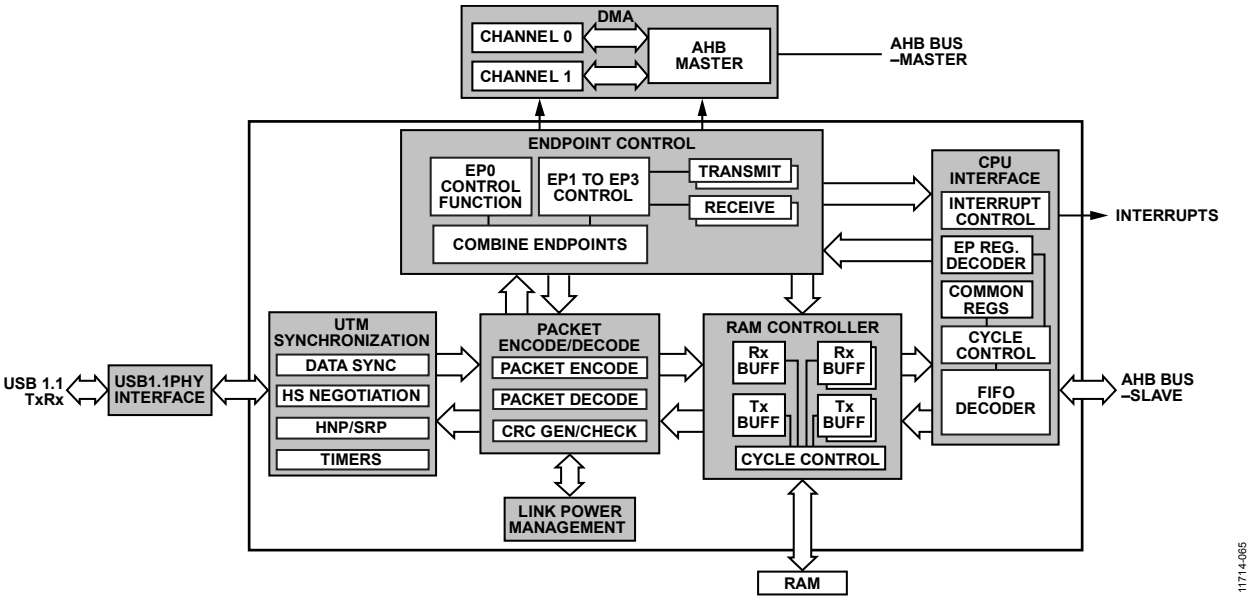


Figure 66. USB Controller Block Diagram

The USB controller has two clock domains: USBPHYCLK and USBCTLCLK. USBPHYCLK is a fixed 60 MHz clock used to clock logic related to the USB PHY, and USBCTLCLK is a minimum 30 MHz clock used for the system interface and packet handling. The system PLL must be enabled for USB transfers, and a HCLK (core) rate of 15 MHz or 16 MHz must be selected during USB transfers. The AHB interfaces work with the USBCTLCLK clock.

Both USBPHYCLK and USBCTLCK are automatically controlled and may be gated off when the USB is in an idle state or not in use (not connected). In these cases, the clocks are enabled automatically when activity is detected on the USB bus.

Use of the USB for transfers is not exclusive and does not require other subsystems to be disabled for performance reasons. Transfers are supported at any time and with all other subsystems active.

UTM Synchronization

The role of the UTM synchronization block is to synchronize signals between the transceiver 60 MHz USBPHYCLK clock domain and the USBCTLCLK clock of the controller.

Packet Encoding/Decoding

The packet encode/decode block generates headers for packets to be transmitted and decodes the headers on received packets. It also generates the CRC for packets to be transmitted and checks the CRC on received packets.

Endpoint Controllers

Two controller state machines are used: one for control transfers over Endpoint 0, and one for bulk/interrupt/isochronous transactions over Endpoint 1 to Endpoint 3.

CPU Interface

The CPU interface allows access to the control/status registers and the FIFOs for each endpoint. It also generates interrupts to the CPU when packets are successfully transmitted or received, and when the core enters suspend mode or resumes from suspend mode. The interface provided is a 32-bit AMBA AHB interface.

RAM Controller

The RAM controller provides an interface to a single block of synchronous single port RAM that is used to buffer packets between the CPU and USB. It takes the FIFO pointers from the endpoint controllers, converts them to address pointers within the RAM block, and generates the RAM access control signals.

DMA Controller

A 2-channel DMA controller is designed for efficient loading/unloading of the endpoint FIFOs. The DMA controller has its own block of control registers and its own interrupt controller. It supports two modes of operation, and each channel can be independently programmed for operating mode.

USB PHY Interface

The USB PHY interface allows the core to operate at full speed.

In Tx operations, the PHY interface provides conversion of 8-bit parallel to serial data, automatic addition of synchronization and end of packet bits, NRZI encoding, and automatic bit stuffing. In Rx operations, this module provides a digital phase-locked loop for receive data, conversion of serial data to 8-bit parallel data, stripping of synchronization and end of packet bits, NRZI decoding, and stuff bit error checking.

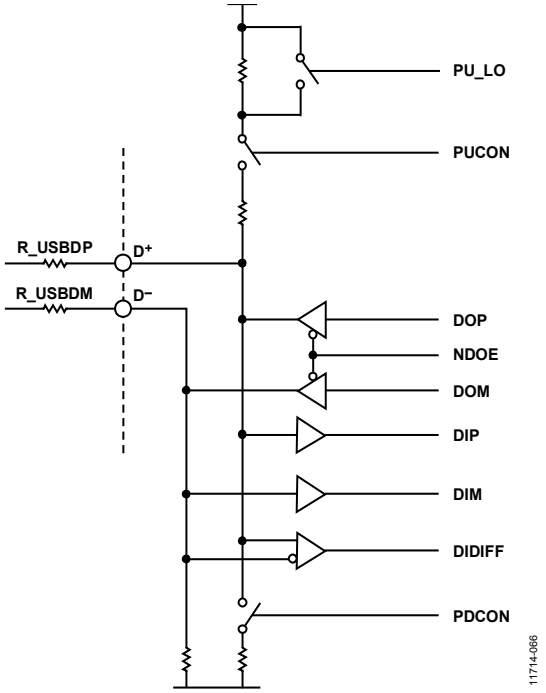


Figure 67. Full Speed USB PHY Diagram

As shown in Figure 67, the DOP and DOM outputs drive the D+ and D− wires through buffers that are enabled when NDOE (active low) is asserted. Likewise, the DIP and DIM inputs are driven by single-ended drivers connected to D+ and D−. The DIDIFF input is driven from a differential driver connected to D+ and D−.

Similarly, the PUCON, PU_LO, and PDCON signals are used to connect a pull-up resistor or pull-down resistor to the D+ wire. A pull-down resistor is connected on the D− wire, and a pull-up resistor is connected on the USB D+ wire. When PUCON is high, the pull-up resistor is connected between the D+ wire and 3.3 V. When PUCON is low, this pull-up resistor is disconnected. Similarly, the pull-down resistor is connected between D+ and ground when PDCON is high, and disconnected when PDCON is low. The PU_LO signal enables the device to select between the higher and lower value pull-up resistors allowed by Section 7.1.5 of the USB 2.0 specification by indicating when the USB bus is idle.

The external termination series resistors, R_USBDP and R_USBDM, are specified at 30 Ω nominal. However, this value may change slightly depending on the real driver output capability. A place holder for these external resistors is sufficient for now. The total impedance looking into the D+ and D− nodes must be between 28 Ω and 44 Ω to achieve optimal reflection reduction, with a typical USB cable characteristic impedance of 45 Ω.

Note that in suspend mode, the drivers are powered down.

Soft Connect/Disconnect

The connection of the USB controller to the USB bus is controlled by software using a feature known as soft connect/disconnect.

When the soft connect/disconnect feature is selected, the PHY can be switched between normal mode and nondriving mode by setting/clearing Bit 6 (SOFTCONN) of the POWER register. When the SOFTCONN bit is set to 1, the PHY is placed in its normal mode

and the D+/D− lines of the USB bus are enabled. At the same time, the USB controller is placed in a powered state, in which it does not respond to any USB signaling, except a USB reset.

When this feature is enabled and the SOFTCONN bit is 0, the PHY is put into nondriving mode, D+ and D− are tristated, and the USB controller appears to other devices on the USB bus as if it has been disconnected.

After a hardware reset, SOFTCONN is cleared to 0. The USB controller therefore appears disconnected until the software has set SOFTCONN to 1. The application software can then choose when to set the PHY into its normal mode. Systems with a lengthy initialization procedure may use this to ensure that initialization is complete, and the system is ready to perform enumeration before connecting to the USB.

When the SOFTCONN bit is set to 1, the software can also simulate a disconnection by clearing this bit to 0.

OPERATION

This section describes how the USB controller operates with regard to the Tx endpoints, Rx endpoints, entry into and exit from suspend mode, and recognition of start of frame (SOF) packets.

IN Transaction

Data for IN transactions is handled through the Tx FIFOs.

When the maximum packet size is set to less than or equal to half the FIFO size, double packet buffering is enabled (if not disabled) for IN transactions. When the maximum packet size is greater than half the FIFO size, single packet buffering is enabled. When double packet buffering is enabled, two data packets can be buffered in the FIFO. When single packet buffering is enabled, only one packet can be buffered, even if the packet is less than half the FIFO size.

Note that the maximum packet size set for any endpoint must not exceed the FIFO size. Also note that the TXMAXP register must not be written to while there is data in the FIFO because unexpected results may occur.

Single Packet Buffering

If the size of the Tx endpoint FIFO is less than twice the maximum packet size for this endpoint (as set in the TXMAXP register), only one packet can be buffered in the FIFO, and single packet buffering is enabled.

As each packet to be sent is loaded into the Tx FIFO, the TXPKTRDY bit in the TXCSR register must be set. If the AUTOSET bit in the TXCSR register is set, the TXPKTRDY bit is automatically set when a maximum sized packet is loaded into the FIFO. For packet sizes less than the maximum and where AUTOSET may not be used, TXPKTRDY must be set manually (that is, by the CPU).

When the TXPKTRDY bit is set, either manually or automatically, the packet is deemed ready to be sent. The NEFIFO bit in the TXCSR register is also set.

When the packet has been successfully sent, both TXPKTRDY and NEFIFO are cleared, and the appropriate Tx endpoint interrupt is generated (if enabled). The next packet can then be loaded into the FIFO.

Double Packet Buffering

Note that double packet buffering is disabled if the corresponding TXDPKTBUFDIS bit of an endpoint is asserted (set to 1) in the TXDPKTBUFDIS register. The default setting for this bit is 0.

If the size of the Tx endpoint FIFO is at least twice the maximum packet size for this endpoint (as set in the TXMAXP register), two packets can be buffered in the FIFO, and double packet buffering is enabled.

As each packet to be sent is loaded into the Tx FIFO, the TXPKTRDY bit in the TXCSR register needs to be set. If the AUTOSET bit in the TXCSR register is set, the TXPKTRDY bit is automatically set when a maximum sized packet is loaded into the FIFO. For packet sizes less than the maximum, and where AUTOSET may not be used, TXPKTRDY must be set manually (that is, by the CPU).

When the TXPKTRDY bit is set, either manually or automatically, the packet is deemed ready to be sent. The NEFIFO bit in the TXCSR register is also set.

After the first packet is loaded, TXPKTRDY is immediately cleared, and an interrupt is generated. A second packet can now be loaded into the Tx FIFO, and TXPKTRDY can be set again (either manually or automatically if the packet is the maximum size). Both packets are now ready to be sent.

After each packet has been successfully sent, TXPKTRDY is cleared, and the appropriate Tx endpoint interrupt is generated (if enabled) to signal that another packet can now be loaded into the Tx FIFO. The state of the NEFIFO bit at this point indicates how many packets can be loaded. If the NEFIFO bit is set, there is another packet in the FIFO, and only one more packet can be loaded. If the NEFIFO bit is clear, there are no packets in the FIFO, and two more packets can be loaded.

OUT Transaction

Data for OUT transactions is handled through the Rx FIFOs.

When the maximum packet size is set to less than or equal to half the FIFO size, double packet buffering is enabled for OUT transactions. When the maximum packet size is greater than half the FIFO size, single packet buffering is enabled. When double packet buffering is enabled, two data packets can be buffered in the FIFO. When single packet buffering is enabled, only one packet can be buffered, even if the packet is less than half the FIFO size.

Note that the maximum packet size must not exceed the FIFO size.

Single Packet Buffering

If the size of the Rx endpoint FIFO is less than twice the maximum packet size for this endpoint (as set in the RXMAXP register), only one data packet can be buffered in the FIFO, and single packet buffering is enabled.

When a packet is received and placed in the Rx FIFO, the RXPKTRDY bit and the FIFOFULL bit in the RXCSR register are set, and the appropriate Rx endpoint is generated (if enabled) to signal that a packet can now be unloaded from the FIFO.

After the packet has been unloaded, the RXPKTRDY bit must be cleared to allow further packets to be received. If the AUTOCLR bit in the RXCSR register is set, and a maximum size packet is unloaded from the FIFO, and the RXPKTRDY bit is cleared automatically. The FIFOFULL bit is also cleared. For packet sizes less than the maximum, RXPKTRDY must be cleared manually (that is, by the CPU, with exceptions; for more information, see the RXPKTRDY, RXCSR, RXCSR1, RXCSR2, and RXCSR3 register descriptions in Table 244, Table 247, Table 270, Table 276, and Table 282, respectively).

Double Packet Buffering

Note that double packet buffering is disabled if the corresponding RXDPKTBUFDIS bit of an endpoint is asserted (equals 1) in the RXDPKTBUFDIS register. The default setting for this bit is 0.

If the size of the Rx endpoint FIFO is at least twice the maximum packet size for the endpoint (as set in the RXMAXP register), two data packets can be buffered, and double packet buffering is enabled.

When the first packet to be received is loaded into the Rx FIFO, the RXPKTRDY bit in the RXCSR register is set, and the appropriate Rx endpoint interrupt is generated (if enabled) to signal that a packet can now be unloaded from the FIFO. Note that the FIFOFULL bit in the RXCSR register is not set at this point; this bit is only set if a second packet is received and loaded into the Rx FIFO.

After each packet has been unloaded, RXPKTRDY must be cleared to allow further packets to be received. If the AUTOCLR bit in the RXCSR register is set and a maximum size packet is unloaded from the FIFO, the RXPKTRDY bit is cleared automatically. For packet sizes less than the maximum, RXPKTRDY must be cleared manually (that is, by the CPU).

If the FIFOFULL bit was set to 1 when RXPKTRDY was cleared, the USB controller first clears the FIFOFULL bit. It then sets RXPKTRDY again to indicate that there is another packet waiting in the FIFO to be unloaded.

Additional Actions

The USB controller responds automatically to certain conditions on the USB bus or actions by the host.

Stall Issued to Control Transfer

The USB controller automatically issues a stall handshake to a control transfer under the following conditions:

- The host sends more data during an OUT data phase of a control transfer than was specified in the device request during the setup phase.
- This condition is detected by the USB controller when the host sends an OUT token (instead of an IN token) after the CPU has unloaded the last OUT packet and set DATAEND.
- The host requests more data during an IN data phase of a control transfer than was specified in the device request during the setup phase.
- This condition is detected by the USB controller when the host sends an IN token (instead of an OUT token) after the CPU has cleared TXPKTRDY and set DATAEND in response to the acknowledge issued by the host to what must have been the last packet.
- The host sends more than MaxP (TXMAXP register, wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9) data with an OUT data token.
- The host sends more than a zero length data packet for the OUT status phase.

Zero-Length OUT Data Packets in Control Transfers

A zero-length OUT data packet is used to indicate the end of a control transfer. In normal operation, such packets must only be received after the entire length of the device request has been transferred (that is, after the CPU has set DATAEND). If, however, the host sends a zero-length OUT data packet before the entire length of device request has been transferred, this signals the premature end of the transfer. In this case, the USB controller automatically flushes any IN token loaded by the CPU that is ready for the data phase from the FIFO and sets SETUPEND.

Suspend

When no activity has occurred on the USB bus for 3 ms, the USB controller enters suspend mode. If the suspend interrupt has been enabled, an interrupt is generated at this time. In suspend mode, the SUSPENDM output goes low (if enabled), which is used to stop USBPHYCLK and put the PHY into suspend mode. In addition, the POWERDWN output is asserted and stops USBCTLCLK and therefore saves power while in this state. POWERDWN then remains asserted until either power is removed from the bus (indicating that the device has been disconnected) or USB resumes signaling or USB reset signaling is detected on the bus.

When resume signaling is detected, the USB controller exits suspend mode and pulls SUSPENDM high. In response, the PHY is taken out of suspend mode. If the resume interrupt is enabled, an interrupt is generated. The CPU can also force the USB controller to exit suspend mode by setting the RESUME bit in the power register. When this bit is set, the USB controller exits suspend mode and drives resume signaling onto the bus. The CPU must clear this bit after 10 ms (a maximum of 15 ms) to end resume signaling.

No resume interrupt is generated when suspend mode is exited by the CPU.

Start of Frame

The USB controller must receive a start-of-frame packet from the host once every millisecond for full speed mode.

When the SOF packet is received, the 11-bit frame number contained in the packet is written into the frame register, and an output pulse, lasting one USBCTLCLK period, is generated on SOF_PULSE. An SOF interrupt is also generated (if enabled in the INTRUSBE register).

After the USB controller starts to receive SOF packets, it expects one every millisecond. If no SOF packet is received after 1.00358 ms (or 125.125 µs), it is assumed that the packet has been lost, and an SOF_PULSE (together with an SOF interrupt, if enabled) is still generated but the frame register is not updated. The USB controller continues to generate a SOF_PULSE every millisecond and resynchronizes these pulses to the received SOF packets when these packets are successfully received again.

USB Reset

When a reset condition is detected on the USB, the device performs the following actions:

- Set FADDR to 0.
- Set index to 0.
- Flush all endpoint FIFOs.
- Clear all control/status registers.
- Enable all endpoint interrupts.
- Generate a reset interrupt.

USB Wake-Up

The USB controller can wake up the system from all low power modes. In the case of the CORE_SLEEP mode, interrupts from the USB and the USB DMA can wake up the core directly provided that the corresponding USB interrupts are enabled in the core. To wake the system up from SYS_SLEEP or hibernate, wake up logic must be activated by writing to the EI2CFG register. Refer to the System Interrupts and Exceptions section for more details.

PROGRAMMING SCHEME

This section describes the actions that the device controlling the USB controller must perform, as well as various aspects of the operation of the USB controller.

USB Interrupt Handling

When the CPU is interrupted with a USB interrupt, it must read the interrupt status register to determine which endpoint(s) caused the interrupt so that it can begin the appropriate interrupt service routine. If multiple endpoints have caused the interrupt, Endpoint 0 is serviced first, followed by the other endpoints.

A flowchart for the USB interrupt service routine is shown in Figure 68.

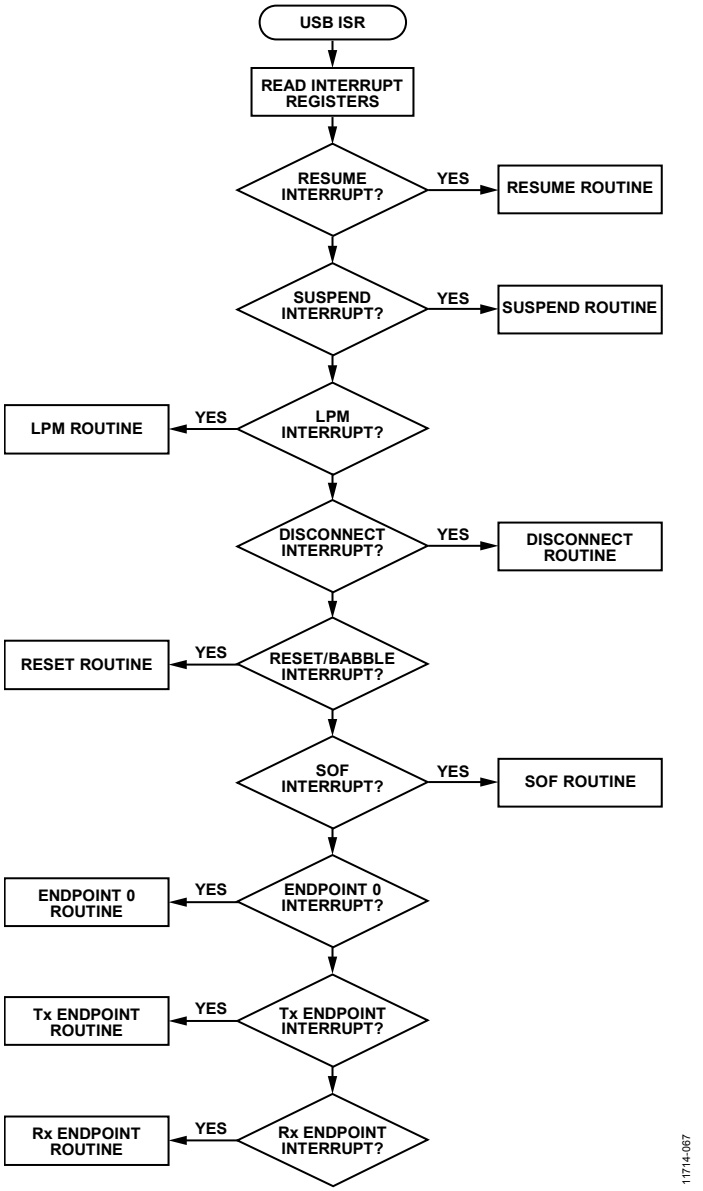


Figure 68. USB Interrupt Service Routine

DMA Controller

The USB DMA master controller supports two DMA modes: DMA Mode 0 and DMA Mode 1.

In addition, the controller can be programmed to conduct transfers on the AHB bus using INCR4, INCR8, and INCR16 4-/8-/16-beat incrementing bursts rather than bursts of an unspecified length.

When operating in DMA Mode 0, the DMA controller can only be programmed to load/unload one packet; therefore, processor intervention is required for each packet transferred over the USB. This mode can be used with any endpoint, whether it uses control, bulk, isochronous, or interrupt transactions (that is, including Endpoint 0).

When operating in DMA Mode 1, the DMA controller can be programmed to load/unload a complete bulk transfer (which can be many packets). After setup, the DMA controller loads/unloads all packets of the transfer, interrupting the processor only when the transfer has completed. DMA Mode 1 can only be used with endpoints that use bulk transactions.

Each channel can be independently programmed for the selected operating mode.

DMA Registers

The DMA controller has one interrupt register that indicates which channels have a pending interrupt and a set of three control registers for each configured channel. These registers are described in detail in Table 285 to Table 291.

Bus Errors

If a bus error occurs while the DMA controller is accessing memory on the AHB, the DMA controller immediately terminates the DMA transfer and interrupts the processor with the bus error (ERR) bit of the DMA_CNLx register set.

Note that the generation of this interrupt is not affected by the setting of the interrupt enable (IE) bit in the DMA_CNLx register. The interrupt is still generated when the IE bit in the DMA_CNLx register = 0.

Transferring Packets

Use of the built-in DMA controller to access the USB controller FIFOs requires both the DMA controller and the endpoint to be appropriately programmed. Many variations are possible. The following sections details the standard setups used for the basic actions of transferring individual packets and multiple packets.

INDIVIDUAL PACKET: RX ENDPOINT

The transfer of individual packets is usually carried out using DMA Mode 0.

For this, the USB controller Rx endpoint must be programmed as follows:

- Set the relevant interrupt enable bit in the INTRRXE register to 1.
- Set the DMAREQEN bit (D13) of the appropriate RXCSR register to 0. Note that there is no need to set the USB controller to support DMA for this operation.

When a packet has been received by the USB controller, it generates the appropriate endpoint interrupt. The processor must then program the selected channel of the DMA controller as follows:

- FADDR: memory address to store packet
- COUNT: size of packet (determined by reading the RXCNTx register)
- CNTL: DMA enable (D0) = 1; direction (D1) = 0; DMA mode (D2) = 0; interrupt enable (D3) = 1; required burst mode (D10 to D9)

The DMA controller then requests the bus mastership and transfers the packet to memory. When it has completed the transfer, it generates a DMA interrupt. The processor must then clear the RXPKTRDY bit in the RXCSR register.

INDIVIDUAL PACKET: TX ENDPOINT

To carry out this operation using DMA Mode 0, a Tx endpoint must be programmed as follows:

- Set the relevant interrupt enable bit in the INTRTXE register to 1.
- Set the DMAREQEN bit (D12) of the appropriate TXCSR register to 0. Note that there is no need to set the USB controller to use DMA for this operation.

When the FIFO in the USB controller becomes available, the USB controller interrupts the processor with the appropriate Tx endpoint interrupt. The processor then programs the DMA controller as follows:

- FADDR: memory address of packet to send
- COUNT: size of packet to be sent
- CNTL: DMA enable (D0) = 1; direction (D1) = 1; DMA mode (D2) = 0; interrupt enable (D3) = 1; required burst mode (D10 to D9)

The DMA controller then requests the bus mastership and transfers the packet to the USB controller FIFO. When it has completed the transfer, it generates a DMA interrupt. The processor then sets the TXPKTRDY bit in the TXCSR register.

MULTIPLE PACKETS: RX ENDPOINT

The transfer of multiple packets is usually carried out using DMA Mode 1.

Where multiple packets are to be received using DMA Mode 1, the DMA controller must be programmed as follows:

- FADDR: memory address of the buffer in which to store transfer.
- COUNT: maximum size of data buffer.
- CNTL: DMA enable (D0) = 1; direction (D1) = 0; DMA mode (D2) = 1; interrupt enable (D3) = 1; required burst mode (D10 to D9).

The Rx endpoint must be programmed as follows:

- Set the relevant interrupt enable bit in the INTRRXE register to 1.
- Set the AUTOCLR (D15), DMAREQEN (D13), and DMAREQMODE (D11) bits of the appropriate RXCSR register to 1.

As each packet is received by the USB controller, the DMA controller requests the bus mastership and transfers the packet to memory. With AUTOCLR set, the USB controller automatically clears the RXPKTRDY bit.

This process continues automatically until the USB controller receives a short packet (one of less than the maximum packet size for the endpoint), signifying the end of the transfer. The short packet is not transferred by the DMA controller; instead, the USB controller interrupts the processor by generating the appropriate endpoint interrupt. The processor can then read the USB controller RXCNT register to see the size of the short packet and either unload it manually or reprogram the DMA controller in Mode 0 to unload the packet.

The DMA controller FADDR register was incremented as the packets were unloaded; therefore, the processor can determine the size of the transfer by comparing the current value of FADDR against the start address of the memory buffer.

Note that if the size of the transfer exceeds the data buffer size, the DMA controller stops unloading the FIFO and interrupts the processor via the DMA interrupt line.

MULTIPLE PACKETS: TX ENDPOINT

To carry out this operation using DMA Mode 1, the DMA controller must be programmed as follows:

- FADDR: memory address of data block to send.
- COUNT: size of data block.
- CNTL: DMA enable (D0) = 1; direction (D1) = 1; DMA mode (D2) = 1; interrupt enable (D3) = 1; required burst mode (D10 to D9).

The Tx endpoint must be programmed as follows:

- Set the relevant interrupt enable bit in the INTRTXE register to 1 (simply so that errors can be detected).
- Set the AUTOSET (D15), DMAREQEN (D12), and DMAREQMODE (D10) bits of the appropriate TXCSR register to 1.

When the FIFO in the USB controller becomes available, the DMA controller requests the bus mastership and transfers a packet to the FIFO. With AUTOSET set, the USB controller automatically sets the TXPKTRDY bit. This process continues until the entire data block has been transferred to the USB controller. The DMA controller then interrupts the processor by asserting the DMA interrupt line. If the last packet to be loaded was less than the maximum packet size for the endpoint, the TXPKTRDY bit is not set for this packet. The processor therefore responds to the DMA interrupt by setting the TXPKTRDY bit to allow the last short packet to be sent. If the last packet to be loaded was of the maximum packet size, the action to take depends on whether the transfer is under the control of an application, such as the mass storage software on a Windows® system that keeps count of the individual packets sent. If the transfer is not under such control, the processor still responds to the DMA interrupt by setting the TXPKTRDY bit. This has the effect of sending a null packet for the receiving software to interpret, indicating the end of the transfer.

USB REGULATOR

The USB regulator takes the 5 V V_{BUS} voltage input and regulates it to the V_{USB} supply.

The V_{USB} supply only supplies the USB PHY; it cannot be used to supply any other section of ADuC M350.

Block Diagram

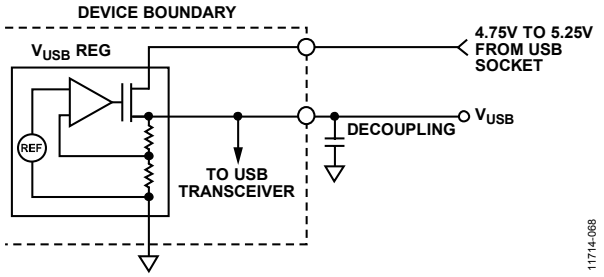


Figure 69. Regulator Block Diagram

USB REGISTER EXCEPTIONS

Only DEVCTL[4:3] = 00 and DEVCTL[4:3] = 11 are valid. The combinations DEVCTL[4:3] = 01 and DEVCTL[4:3] = 10 are reserved.

FULL SPEED USB DEVICE CONTROLLER (FSUSB) MEMORY MAPPED REGISTERS

FSUSB Register Map

Table 231. FSUSB Register Summary

Address	Name	Description	Reset	RW
0x400A0000	FADDR	Device address	0x00	R/W
0x400A0001	POWER	Power and device control	0x20	R/W
0x400A0002	INTRTX	Transmit interrupt	0x0000	R/NW
0x400A0004	INTRRX	Receive interrupts	0x0000	R/NW
0x400A0006	INTRTXE	Transmit interrupt enable	0x000F	R/W
0x400A0008	INTRRXE	Receive interrupt enable	0x000E	R/W
0x400A000A	INTRUSB	USB interrupts	0x00	R/NW
0x400A000B	INTRUSBE	USB interrupt enable	0x06	R/W
0x400A000C	FRAME	Frame number	0x0000	R/NW
0x400A000E	INDEX	Index register for selecting the endpoint status and control registers	0x00	R/W
0x400A000F	TESTMODE	Enables USB 2.0 Test Modes	0x00	R/W
0x400A0010	TXMAXP	Transmit maximum packet length with Index 1 to Index 3	0x0000	R/W
0x400A0012	CSR0	Transmit configuration and status with Index 1 to Index 3	0x0000	R/W
0x400A0012	TXCSR	Control status register of EP0 when index register set to 0	0x0000	R/W
0x400A0014	RXMAXP	Maximum packet size for Rx endpoint with Index 1 to Index 3	0x0000	R/W
0x400A0016	RXCSR	Control status register for Rx endpoint with Index 1 to Index 3	0x0000	R/W
0x400A0018	CNT0	Number of received bytes for Endpoint 0 when Index Register Set to 0	0x0000	R/NW
0x400A0018	RXCNT	Number of byte received with Index 1 to Index 3	0x0000	R/NW
0x400A001F	CFGDATA	Configuration information when index register set to 0	0x02	R/NW
0x400A001F	FIFOSIZE	Tx/Rx FIFO size of endpoint with Index 1 to Index 3	0x00	R/NW
0x400A0020	FIFO0	FIFO data for Endpoint n	0x0000000X	R/W
0x400A0024	FIFO1	FIFO data for Endpoint 1	0xXXXXXXXX	RW
0x400A0028	FIFO2	FIFO data for Endpoint 2	0xXXXXXXXX	RW
0x400A002C	FIFO3	FIFO data for Endpoint 3	0xXXXXXXXX	RW
0x400A0060	DEVCTL	Device control	0x80	R/W
0x400A0061	MISC	Miscellaneous	0x00	RW
0x400A006C	HWVERS	Hardware version	0x0800	R/NW
0x400A0078	EPINFO	Endpoint info	0x33	R/NW
0x400A0079	RAMINFO	Ram information	0x29	R/NW
0x400A007A	LINKINFO	Programmable delay values	0x5C	R/W
0x400A007D	FS_EOF1	Time buffer available on full-speed transaction	0x77	R/W
0x400A007F	SOFT_RST	Software reset	0x00	R/W
0x400A0102	CSR0	Control status register for Endpoint 0	0x0000	R/W
0x400A0108	CNT0	Number of received bytes for Endpoint 0	0x0000	R/NW
0x400A010F	CFGDATA	Configuration information for Endpoint 0	0x02	R/NW
0x400A0110	TXMAXP1	Maximum packet size for Tx Endpoint 1	0x0000	R/W
0x400A0112	TXCSR1	Control status register for Tx Endpoint 1	0x0000	R/W
0x400A0114	RXMAXP1	Maximum packet size for Rx Endpoint 1	0x0000	R/W
0x400A0116	RXCSR1	Control status register for Rx Endpoint 1	0x0000	R/W
0x400A0118	RXCNT1	Number of byte received for Rx Endpoint 1	0x0000	R/NW
0x400A011F	FIFOSIZE1	Tx/Rx FIFO size for Endpoint 1	0xXX	R/NW
0x400A0120	TXMAXP2	Maximum packet size for Tx Endpoint 2	0x0000	R/W
0x400A0122	TXCSR2	Control status register for Tx Endpoint 2	0x0000	R/W
0x400A0124	RXMAXP2	Maximum packet size for Rx Endpoint 2	0x0000	R/W
0x400A0126	RXCSR2	Control status register for Rx Endpoint 2	0x0000	R/W
0x400A0128	RXCNT2	Number of byte received for Rx Endpoint 2	0x0000	R/NW
0x400A012F	FIFOSIZE2	Tx/Rx FIFO size for Endpoint 2	0xXX	R/NW
0x400A0130	TXMAXP3	Maximum packet size for Tx Endpoint 3	0x0000	R/W
0x400A0132	TXCSR3	Control status register for Tx Endpoint 3	0x0000	R/W

Address	Name	Description	Reset	RW
0x400A0134	RXMAXP3	Maximum packet size for Rx Endpoint 3	0x0000	R/W
0x400A0136	RXCSR3	Control status register for Rx Endpoint 3	0x0000	R/W
0x400A0138	RXCNT3	Number of byte received for Rx Endpoint 3	0x0000	R/NW
0x400A013F	FIFOSIZE3	Tx/Rx FIFO size for Endpoint 3	0xXX	R/NW
0x400A0200	DMA_IRQ	DMA interrupt register	0x00	R/NW
0x400A0204	DMA_CTL0	DMA control for Channel n	0x0000	R/W
0x400A0208	DMA_ADDR0	DMA address for Channel n	0x00000000	R/WE
0x400A020C	DMA_CNT0	DMA count for Channel n	0x00000000	R/W
0x400A0214	DMA_CTL1	DMA control for Channel 1	0x0000	RW
0x400A0218	DMA_ADDR1	DMA address for Channel 1	0x00000000	RW
0x400A021C	DMA_CNT1	DMA count for Channel 1	0x00000000	RW
0x400A0340	RXDPKTBUFDIS	Rx double packet buffer disable for Endpoint 1 to Endpoint 3	0x0000	R/W
0x400A0342	TXDPKTBUFDIS	Tx double packet buffer disable for Endpoint 1 to Endpoint 3	0x0000	R/W
0x400A0344	CT_UCH	Chirp timeout	0x4074	R/W
0x400A0360	LPM_ATTR	LPM attribute	0x0000	R/W
0x400A0362	LPM_CTL	LPM control	0x00	R/W
0x400A0363	LPM_IEN	LPM interrupt enable	0x00	R/W
0x400A0364	LPM_IRQ	LPM interrupt	0x00	R/W
0x400A0397	RESERVED	Reserved.		
0x400A039C	USBPHYCTL	USB PHY control	0x0000	R/W
0x400A039E	USBPHYSTAT	USB PHY status	0x00XX	R/NW
0x400A03B0	RAM_ADDR	RAM address	0x00000000	R/W
0x400A03B4	RAM_DATA	RAM data	0x0000000X	R/W

Device Address Register

Address: 0x400A0000, Reset: 0x00, Name: FADDR

FADDR is an 8-bit register that is written with the 7-bit address of the peripheral part of the transaction.

This register is written with the address received through a SET_ADDRESS command, which is then used for decoding the function address in subsequent token packets.

Table 232. Bit Descriptions for FADDR

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved.	0x0	R/W
[6:0]	FUNCADDR	The function address. This register is reset by an USB reset.	0x0	R/W

Power and Device Control Register

Address: 0x400A0001, Reset: 0x20, Name: POWER

POWER is an 8-bit register that is used for controlling suspend and resume signaling, and some basic operational aspects.

Table 233. Bit Descriptions for POWER

Bits	Bit Name	Description	Reset	Access
7	ISOUPDT	ISO update. This bit is used to delay sending isochronous packets until SOF is received after TXPKTRDY is set. If an IN token is received before a SOF token, a zero length data packet is sent.	0x0	R/W
6	SOFTCONN	Enable the D± termination resistors. The D± lines default to disconnected. Setting this bit enables the D± termination resistors. This bit is automatically set when the SESSION bit in DEVCTL is written with 1. 0: disable the D± termination resistors. 1: enable the D± termination resistors.	0x0	R/W
5	RESERVED	Reserved.	0x1	R/NW
4	RESERVED	Reserved.	0x0	R/NW
3	RESET	Reset. This bit indicates reset has been detected on the USB bus. This bit is read only.	0x0	R/NW
2	RESUME	Assert resume signaling. This bit causes the controller to assert resume signaling. The CPU clears this bit after 10 ms to 15 ms.	0x0	R/W

Bits	Bit Name	Description	Reset	Access
1	SUSPEND	Suspend mode. This bit is set upon entry to suspend mode and cleared when the CPU reads the interrupt register. This bit is automatically cleared if the resume bit is set.	0x0	R/W
0	SUSEN	Enable UTMI SUSPENDM output. Setting this bit enables the UTMI SUSPENDM output.	0x0	R/W

Transmit Interrupt Register

Address: 0x400A0002, Reset: 0x0000, Name: INTRTX

INTRTX is a 16-bit read only register that indicates which interrupts are currently active for Endpoint 0 and the Tx Endpoint 1 to Endpoint 3. Note that all active interrupts are cleared when this register is read.

Table 234. Bit Descriptions for INTRTX

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	EP3_TX	Tx Endpoint 3 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC
2	EP2_TX	Tx Endpoint 2 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC
1	EP1_TX	Tx Endpoint 1 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC
0	EP0	Endpoint 0 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC

Receive Interrupts Register

Address: 0x400A0004, Reset: 0x0000, Name: INTRRX

INTRRX is a 16-bit read-only register that indicates which of the interrupts for Rx Endpoint 1 to Endpoint 3 are currently active. Note that all active interrupts are cleared when this register is read.

Table 235. Bit Descriptions for INTRRX

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	EP3_RX	RX Endpoint 3 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC
2	EP2_RX	RX Endpoint 2 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC
1	EP1_RX	RX Endpoint 1 interrupt status. 0: interrupt inactive. 1: interrupt active.	0x0	RC
0	RESERVED	Reserved.	0x0	R

Transmit Interrupt Enable Register

Address: 0x400A0006, Reset: 0x000F, Name: INTRTXE

This register enables the corresponding endpoint transmit interrupt.

This register is reset by a USB reset.

Table 236. Bit Descriptions for INTRTXE

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R/NW
3	EP3_TXE	Tx interrupt enable for Endpoint 3. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W
2	EP2_TXE	Tx interrupt enable for Endpoint 2. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W
1	EP1_TXE	Tx interrupt enable for Endpoint 1. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W
0	EP0E	Interrupt enable for Endpoint 0. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W

Receive Interrupt Enable Register

Address: 0x400A0008, Reset: 0x000E, Name: INTRRXE

This register enables the corresponding endpoint receive interrupt.

This register is reset by a USB reset.

Table 237. Bit Descriptions for INTRRXE

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	EP3_RXE	Rx interrupt enable for Endpoint 3. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W
2	EP2_RXE	Rx interrupt enable for Endpoint 2. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W
1	EP1_RXE	Rx interrupt enable for Endpoint 1. 0: interrupt not enabled. 1: interrupt enabled.	0x1	R/W
0	RESERVED	Reserved.	0x0	R/NW

USB Interrupts Register

Address: 0x400A000A, Reset: 0x00, Name: INTRUSB

INTRUSB is an 8-bit read only register that indicates which USB interrupts are currently active. All active interrupts are cleared when this register is read.

Table 238. Bit Descriptions for INTRUSB

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	RC/NW
5	DISCON	Interrupt status of device disconnect. Set when a session ends. 0: no disconnection interrupt detected. 1: disconnection interrupt detected.	0x0	RC/NW
4	RESERVED	Reserved.	0x0	RC/NW

Bits	Bit Name	Description	Reset	Access
3	SOF	Interrupt status of start of frame. Set when a new frame starts. 0: no start of frame interrupt detected. 1: start of frame interrupt detected.	0x0	RC/NW
2	RST	Interrupt status of reset detection. This bit is set when reset signaling is detected on the bus. 0: no reset interrupt detected. 1: reset interrupt detected.	0x0	RC/NW
1	RESUME	Interrupt status of resume signaling detection. Set when resume signaling is detected on the bus while in suspend mode. 0: no resume interrupt detected. 1: resume interrupt detected.	0x0	RC/NW
0	SUSPEND	Interrupt status of suspend signaling detection. Set when suspend signaling is detected on the bus. 0: no suspend interrupt detected. 1: suspend interrupt detected.	0x0	RC/NW

USB Interrupt Enable Register

Address: 0x400A000B, Reset: 0x06, Name: INTRUSBE

INTRUSBE is an 8-bit register that provides interrupt enable bits for each of the interrupts in INTRUSB.

Table 239. Bit Descriptions for INTRUSBE

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	R/NW
5	DISCON	Enable disconnection detection interrupt. 0: disconnection detection interrupt not enabled. 1: disconnection detection interrupt enabled.	0x0	R/W
4	RESERVED	Reserved.	0x0	R/NW
3	SOF	Enable start of frame detection interrupt. 0: start of frame detection interrupt not enabled. 1: start of frame detection interrupt enabled.	0x0	R/W
2	RST	Enable reset detection interrupt. 0: reset detection interrupt not enabled. 1: reset detection interrupt enabled.	0x1	R/W
1	RESUME	Enable resume detection interrupt. 0: resume detection interrupt not enabled. 1: resume detection interrupt enabled.	0x1	R/W
0	SUSPEND	Enable suspend detection interrupt. 0: suspend detection interrupt not enabled. 1: suspend detection interrupt enabled.	0x0	R/W

Frame Number Register

Address: 0x400A000C, Reset: 0x0000, Name: FRAME

Frame is a 16-bit read only register that holds the last received frame number.

Table 240. Bit Descriptions for FRAME

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	FRAME_NUMBER	Frame number. This field holds the current frame number. This field is reset by a USB reset.	0x0	R/NW

Index Register for Selecting the Endpoint Status and Control Registers

Address: 0x400A000E, Reset: 0x00, Name: INDEX

Each Tx endpoint and each Rx endpoint have their own set of control/status registers located between 400A0100h and 400A01FFh. In addition, one set of Tx control/status registers and one set of Rx control/status registers appear at 400A0010h to 400A0019h. Index is a 4-bit register that determines which endpoint control/status registers are accessed.

Before accessing the control/status registers of an endpoint at 400A0010h to 400A0019h, the endpoint number must be written to the index register to ensure that the correct control/status registers appear in the memory map.

Table 241. Bit Descriptions for INDEX

Bits	Bit Name	Description	Reset	Access
[7:4]	RESERVED	Reserved.	0x0	R/W
[3:0]	INDEX	Select endpoint. Currently selected endpoint. Valid values for this register are 0 to 3.	0x0	R/W

Enables USB 2.0 Test Modes Register

Address: 0x400A000F, Reset: 0x00, Name: TESTMODE

This register is used for the required USB 2.0 test modes by SET FEATURE: TESTMODE command. It is not used in normal operation.

Note: Only one bit can be set at any time.

Table 242. Bit Descriptions for TESTMODE

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved.	0x0	R/W
6	FIFOACCESS	FIFO access. This bit is set to transfer the contents of the Endpoint 0 Tx FIFO into the Endpoint 0 Rx FIFO. This bit is cleared automatically. 0: no action. 1: FIFO access enabled.	0x0	R/W1A
5	FORCEFS	Force FS. This bit is set to force the controller into full speed mode when a USB reset is received. 0: no action. 1: force the device into full speed mode.	0x0	R/W
[4:0]	RESERVED	Reserved.	0x0	R/W

Transmit Maximum Packet Length with the Index 1 to Index 3 Register

Address: 0x400A0010, Reset: 0x0000, Name: TXMAXP

The TXMAXP register defines the maximum amount of data that can be transferred through the selected TX endpoint in a single operation. There is a TXMAXP register for each Tx endpoint of 1 to 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Tx endpoint, and must not exceed half the FIFO size if double buffering is required.

If this register is changed after packets have been sent from the endpoint, the Tx endpoint FIFO must be completely flushed (using the FLUSHFIFO bit in TXCSR register) after writing the new value to this register.

Note that TXMAXP must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 243. Bit Descriptions for TXMAXP

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that can be transferred per transaction.	0x0	R/W

Control Status Register of EP0 When Index Register Set to 0

Address: 0x400A0012, Reset: 0x0000, Name: CSR0

CSR0 is a 16-bit register that provides control and status bits for Endpoint 0.

Table 244. Bit Descriptions for CSR0

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	FLUSHFIFO	Flush FIFO. The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit (Bit 1 and Bit 0) is cleared. Note that FLUSHFIFO must only be used when TXPKTRDY/RXPKTRDY is set. At other times, it may cause data to be corrupted.	0x0	RW
7	SSETUPEND	Serviced setup end. The CPU writes a 1 to this bit to clear the SETUPEND bit. It is cleared automatically.	0x0	RW
6	SRXPKTRDY	Serviced RXPKTRDY. The CPU writes a 1 to this bit to clear the RXPKTRDY bit. It is cleared automatically.	0x0	RW
5	SENDSTALL	Send stall. The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshake is transmitted and then this bit is cleared automatically.	0x0	RW
4	SETUPEND	Setup end. This bit is set when a control transaction ends before the DATAEND bit has been set. An interrupt is generated, and the FIFO is flushed at this time. The bit is cleared by the CPU writing a 1 to the SSETUPEND bit.	0x0	RW
3	DATAEND	Data end. The CPU sets this bit when setting TXPKTRDY for the last data packet, when clearing RXPKTRDY after unloading the last data packet, or when setting TXPKTRDY for a zero length data packet. It is cleared automatically.	0x0	RW
2	SENTSTALL	Sent stall. This bit is set when a STALL handshake is transmitted. The CPU must clear this bit.	0x0	RW
1	TXPKTRDY	Tx packet ready. The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).	0x0	RW
0	RXPKTRDY	Rx packet ready. This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The CPU clears this bit by setting the SRXPKTRDY bit.	0x0	RW

Transmit Configuration and Status with the Index 1 to Index 3 Register

Address: 0x400A0012, Reset: 0x0000, Name: TXCSR

TXCSR is a 16-bit register that provides control and status bits for transfers through the currently selected Tx endpoint. There is a TXCSR register for each configured Tx Endpoint 1 to Endpoint 3 (not including Endpoint 0).

Table 245. Bit Descriptions for TXCSR

Bits	Bit Name	Description	Reset	Access
15	AUTOSET	If this bit is set, TXPKTRDY is automatically set when the maximum data packet size TXMAXP is loaded into the Tx FIFO. TXMAXP must be a word (4-byte) multiple. If a packet less than the maximum packet size is loaded, the TXPKTRDY bit must be set manually.	0x0	RW
14	ISO	This bit must be set to enable the TX endpoint for isochronous transfers. This bit must be clear for bulk or interrupt endpoints.	0x0	RW
13	RESERVED	Reserved.	0x0	R
12	DMAREQEN	Set this bit to enable DMA requests for this Tx endpoint.	0x0	RW
11	FRCDATATGL	Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an acknowledge was received. This can be used by interrupt Tx endpoints that are used to communicate rate feedback for isochronous endpoints.	0x0	RW
10	DMAREQMODE	Set this bit to select DMA Request Mode 1, or clear this bit to select DMA Request Mode 0. This bit must not be cleared the cycle before or the same cycle that DMAREQEN is cleared. In DMA Request Mode 0, the DMA is programmed to load one packet at a time. Processor intervention is required for each packet. DMA Request Mode 1 can be used with bulk endpoints to transmit multiple packets without CPU intervention. This bit must not be cleared either before or in the same cycle as the DMAREQEN bit is cleared.	0x0	RW
[9:7]	RESERVED	Reserved.	0x0	R
6	CLRDATATGL	The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.	0x0	RW

Bits	Bit Name	Description	Reset	Access
5	SENTSTALL	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. The CPU must clear this bit.	0x0	RW
4	SENDSTALL	The CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the STALL condition. This bit has no effect for isochronous transfers.	0x0	RW
3	FLUSHFIFO	The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer is reset, the TXPKTRDY bit is cleared, and an interrupt is generated. This bit can be set simultaneously with TXPKTRDY to abort the packet that is currently being loaded into the FIFO. FLUSHFIFO must only be used when the TXPKTRDY is set. At other times, it can cause data to be corrupted. Also, note that if the FIFO is double buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.	0x0	RW
2	URUNERR	This bit is set if an IN token is received when TXPKTRDY is not set. The CPU must clear this bit.	0x0	RW
1	NEFIFO	This bit is set when there is at least one packet in the Tx FIFO.	0x0	RW

Maximum Packet Size for Rx Endpoint with Index 1 to Index 3 Register

Address: 0x400A0014, Reset: 0x0000, Name: RXMAXP

The RXMAXP register defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is an RXMAXP register for each Rx Endpoint 1to Rx Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch may cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Rx endpoint, and must not exceed half the FIFO size if double buffering is required.

RXMAXP must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 246. Bit Descriptions for RXMAXP

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that may be transferred per transaction.	0x0	R/W

Control Status Register for Rx Endpoint with Index 1 to Index 3

Address: 0x400A0016, Reset: 0x0000, Name: RXCSR

RXCSR is a 16-bit register that provides control and status bits for transfers through the currently selected Rx endpoint. There is a RXCSR register for each configured Rx Endpoint 1to Rx Endpoint 3 (not including Endpoint 0).

Table 247. Bit Descriptions for RXCSR

Bits	Bit Name	Description			Reset	Access
15	AUTOCLR	If the CPU sets this bit, then the RXPKTRDY bit is automatically cleared when a packet of RXMAXP bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY must be cleared automatically. When using the DMA to unload the Rx FIFO, data is read from the Rx FIFO in 4-byte chunks regardless of RXMAXP. Therefore, the RXPKTRDY bit is cleared as follows:			0x0	R/W
		(RXMAXP%4)	**Actual Bytes Read**	**Packet Sizes that Clear RXPKTRDY**		
		0	RXMAXP	RXMAXP, RXMAXP − 1, RXMAXP − 2, RXMAXP − 3		
		3	RXMAXP + 1	RXMAXP, RXMAXP − 1, RXMAXP − 2		
		2	RXMAXP + 2	RXMAXP, RXMAXP − 1		
		1	RXMAXP + 3	RXMAXP		
14	ISO	This bit must be set for isochronous transfers and cleared for bulk or interrupt transfers.			0x0	R/W
13	DMAREQEN	The CPU must set this bit to enable DMA requests for the Rx endpoint.			0x0	R/W
12	PIDERR	For isochronous transactions, this bit indicates a PID error in the received packet.			0x0	R/W
11	DMAREQMODE	The CPU must set this bit to select DMA Request Mode 1 or clear this bit to select DMA Mode 0. In DMA Mode 0, CPU intervention is required for every packet that is received. For bulk transactions, DMA Mode 1 can be used to transfer multiple RXMAXP packets.			0x0	R/W
[10:8]	RESERVED	Reserved.			0x0	R/W
7	CLRDATATGL	If this bit is set, the data toggle bit is cleared.			0x0	R/W
6	SENTSTALL	This bit is set when a STALL handshake is transmitted. The CPU must clear this bit.			0x0	R/W
5	SENDSTALL	This bit must be set to send a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.			0x0	R/W
4	FLUSHFIFO	The CPU must write a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO pointer is cleared and the RXPKTRDY bit is cleared. FLUSHFIFO must only be used when RXPKTRDY is set. At other times, it may cause data to be corrupted. If the FIFO is double buffered, this bit may need to be set twice to completely clear the FIFO.			0x0	R/W
3	DATAERR	In ISO mode, this bit is set when RXPKTRDY is set if the data packet has a CRC or bit stuff error. It is cleared when RXPKTRDY is cleared. This bit is always zero for bulk endpoints.			0x0	R/W
2	ORUNERR	Overrun error. This bit is set if an OUT packet cannot be loaded into the Rx FIFO. The CPU must clear this bit. This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, this bit always returns zero.			0x0	R/W
1	FIFOFULL	This bit is set when no more packets can be loaded into the Rx FIFO.			0x0	R/W
0	RXPKTRDY	This bit is set when a data packet has been received. The CPU must clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when this bit is set.			0x0	R/W

Number of Received Bytes for Endpoint 0 When Index Register Set to 0

Address: 0x400A0018, Reset: 0x0000, Name: CNT0

CNT0 is a 7-bit read only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RXPKTRDY (Bit 0 of RXCSR) is set.

Table 248. Bit Descriptions for CNT0

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	R
[6:0]	CNT0	Rx Count 0.	0x0	R

Number of Byte Received with Index 1 to Index 3 Register

Address: 0x400A0018, Reset: 0x0000, Name: RXCNT

RXCNT is a 14-bit read only register that holds the number of data bytes in the packet currently in line to be read from the Rx FIFO.

Note that the value returned changes as the FIFO is unloaded and is only valid while RXPKTRDY (Bit 0 of RXCSR) is set.

Table 249. Bit Descriptions for RXCNT

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R
[13:0]	RXCNT	Rx count.	0x0	R

Configuration Information When Index Register Set to 0

Address: 0x400A001F, Reset: 0x02, Name: CFGDATA

CFGDATA is an 8-bit read only register that returns information about the selected core configuration with the index register set to 0.

Table 250. Bit Descriptions for CFGDATA

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	R0/NW
5	BIGEND	Big endian configuration. Always 0. Indicates little endian ordering. 1: big endian configuration.	0x0	R0/NW
[4:2]	RESERVED	Reserved.	0x0	R1/NW
1	SOFTCON	Soft connect enable. D1 SOFTCON is always 1. Indicates soft connect/disconnect enabled. 1: Soft connect is enabled.	0x1	R1/NW
0	UTMIWID	UTMI data width. Always 0 indicates, 8-bit UTMI data width. 0: 8-bit UTMI data width. 1: 16-bit UTMI data width.	0x0	R0/W

Tx/Rx FIFO Size of Endpoint with Index 1 to Index 3 Register

Address: 0x400A001F, Reset: 0x00, Name: FIFOSIZE

FIFOSIZE is an 8-bit read only register that returns the sizes of the FIFOs associated with the selected additional Tx/Rx endpoints. The lower nibble encodes the size of the selected Tx endpoint FIFO; the upper nibble encodes the size of the selected Rx endpoint FIFO. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).

Note that the register only has this interpretation when the index register is set to select one of Endpoint 1 to Endpoint 3.

- Endpoint 1 Tx 7 (128-byte), Rx 7 (128-byte).
- Endpoint 2 Tx 8 (256-byte), Rx 8 (256-byte).
- Endpoint 3 Tx 9 (512-byte), Rx 9 (512-byte).

Table 251. Bit Descriptions for FIFOSIZE

Bits	Bit Name	Description	Reset	Access
[7:4]	RXFIFOSZ	Rx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes). A value of 0 for nonexistent endpoint. 0xF if Tx and Rx sharing same FIFO.	0x0	R/NW
[3:0]	TXFIFOSZ	Tx FIFO size. Values of 3 to13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes). A value of 0 for nonexistent endpoint.	0x0	R/NW

FIFO Data for Endpoint 0 Register

Address: 0x400A0020, Reset: 0x0000000X, Name: FIFO0

Table 252. Bit Descriptions for FIFO0

Bits	Bit Name	Description	Reset	Access
[31:0]	FIFO_DATA	Writes to this register go to the endpoint Tx FIFO. Reads from this register come from the endpoint Rx FIFO. FIFO access to/from this register can be 32-bit, 16-bit, or 8-bit. The accesses must be the same size so the data consistently remains byte, half, or word aligned. The last access can be a different size to complete an odd byte or half word transfer because the alignment no longer needs to be maintained after the last access.	0xx	R/W

FIFO Data for Endpoint 1 Register

Address: 0x400A0024, Reset: 0x0000000X, Name: FIFO1

Table 253. Bit Descriptions for FIFO1

Bits	Bit Name	Description	Reset	Access
[31:0]	FIFO_DATA	Writes to this register go to the endpoint Tx FIFO. Reads from this register come from the endpoint Rx FIFO. FIFO access to/from this register can be 32-bit, 16-bit, or 8-bit. The accesses must be the same size so the data consistently remains byte, half, or word aligned. The last access can be a different size to complete an odd byte or half word transfer because the alignment no longer needs to be maintained after the last access.	0xx	R/W

FIFO Data for Endpoint 2 Register

Address: 0x400A0028, Reset: 0x0000000X, Name: FIFO2

Table 254. Bit Descriptions for FIFO2

Bits	Bit Name	Description	Reset	Access
[31:0]	FIFO_DATA	Writes to this register go to the endpoint Tx FIFO. Reads from this register come from the endpoint Rx FIFO. FIFO access to/from this register can be 32-bit, 16-bit, or 8-bit. The accesses must be the same size so the data consistently remains byte, half, or word aligned. The last access can be a different size to complete an odd byte or half word transfer because the alignment no longer needs to be maintained after the last access.	0xx	R/W

FIFO Data for Endpoint 3 Register

Address: 0x400A002C, Reset: 0x0000000X, Name: FIFO3

Table 255. Bit Descriptions for FIFO3

Bits	Bit Name	Description	Reset	Access
[31:0]	FIFO_DATA	Writes to this register go to the endpoint Tx FIFO. Reads from this register come from the endpoint Rx FIFO. FIFO access to/from this register can be 32-bit, 16-bit, or 8-bit. The accesses must be the same size so the data consistently remains byte, half, or word aligned. The last access can be a different size to complete an odd byte or half word transfer because the alignment no longer needs to be maintained after the last access.	0xx	R/W

Device Control Register

Address: 0x400A0060, Reset: 0x80, Name: DEVCTL

DEVCTL is an 8-bit register that is for controlling and monitoring the USB VBUS line. If the PHY is suspended no PHY clock (USBPHYCLK) is received and the VBus is not sampled.

Table 256. Bit Descriptions for DEVCTL

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved.	0x1	R/NW
[6:5]	RESERVED	Reserved.	0x0	R/NW
[4:3]	VBUS	VBUS level. These read only bits encode the current VBus level as follows:. 00: below session end. 01: reserved. 10: reserved. 11: above VBUSVALID.	0x0	R/NW
[2:1]	RESERVED	Reserved.	0x0	R/NW
0	SESSION	This bit is set/cleared by the controller when a session starts/ends. It is also set by the CPU to initiate the session request protocol. When the core is in suspend mode, the bit can be cleared by the CPU to perform a software disconnect. Clearing this bit when the core is not suspended results in undefined behavior.	0x0	R/W

Miscellaneous Register

Address: 0x400A0061, Reset: 0x00, Name: MISC

This MISC register is an 8-bit register that contains various common configuration bits. These bits include the Rx/Tx early DMA enable bits.

Table 257. Bit Descriptions for MISC

Bits	Bit Name	Description	Reset	Access
[7:2]	RESERVED	Reserved.	0x0	R
1	TXEDMA	DMA_REQ for all Tx endpoints. 0: DMA_REQ for all Rx endpoints are deasserted when RXMAXP bytes are read to an endpoint. This is late mode.		
		1: DMA_REQ for all Rx endpoints are deasserted when RXMAXP – 8 bytes are read from an endpoint. This is early mode.	0x2	R/W
0	RXEDMA	DMA_REQ for all Rx endpoints. 0: DMA_REQ for all Tx endpoints are deasserted when TXMAXP bytes are written to an endpoint. This is late mode. 1: DMA_REQ for all Tx endpoints are deasserted when TXMAXP – 8 bytes have been written to an endpoint. This is early mode.	0x0	RW

Hardware Version Register

Address: 0x400A006C, Reset: 0x0800, Name: HWVERS

HWVERS register is a 16-bit read only register that returns information about the version of the RTL from which the core hardware was generated, in particular the RTL version number (vxx.yyy).

Table 258. Bit Descriptions for HWVERS

Bits	Bit Name	Description	Reset	Access
15	RC	Release candidate. Set to 1 if RTL used from a release candidate rather than from a full release of the core. 0: official release version. 1: release candidate.	0x0	R/W
[14:10]	MAJOR	Major version number. Major version number (Range: 0 to 31).	0x2	R/W
[9:0]	MINOR	Minor version number. Minor version number (Range: 0 to 999).	0x0	R/W

Endpoint Info Register

Address: 0x400A0078, Reset: 0x33, Name: EPINFO

This 8-bit read only register allows readback of the number of Tx and Rx endpoints included in the design (except Endpoint 0).

Table 259. Bit Descriptions for EPINFO

Bits	Bit Name	Description	Reset	Access
[7:4]	RXEP	Number of Rx endpoints excluding EP0. The number of Rx endpoints implemented in the design.	0x3	R/NW
[3:0]	TXEP	Number of Tx endpoints excluding EP0. The number of Tx endpoints implemented in the design.	0x3	R/NW

RAM Information Register

Address: 0x400A0079, Reset: 0x29, Name: RAMINFO

This 8-bit read only register provides information about the width of the RAM.

Table 260. Bit Descriptions for RAMINFO

Bits	Bit Name	Description	Reset	Access
[7:4]	DMACHANS	Number of DMA channels. The number of DMA channels implemented in the design.	0x2	R/NW
[3:0]	RAMBITS	Number of RAM address bits. The FIFO ram is 32-bits wide. The number of bytes in the FIFO ram is $2^{(RAMBITS + 2)}$.	0x9	R/NW

Programmable Delay Values Register

Address: 0x400A007A, Reset: 0x5C, Name: LINKINFO

This 8-bit register allows some delays to be specified.

Table 261. Bit Descriptions for LINKINFO

Bits	Bit Name	Description	Reset	Access
[7:4]	WTCON	Sets the wait to be applied to allow for the users connect/disconnect filter in units of 533.3 ns. The default settings corresponds to 2.667 µs.	0x5	R/W
[3:0]	WTID	Sets the delay to be applied from IDPULLUP being asserted to IDDIG being considered valid in units of 4.3690 ms. The default corresponds to 52.43 ms.	0xc	R/W

Time Buffer Available on Full-Speed Transaction Register

Address: 0x400A007D, Reset: 0x77, Name: FS_EOF1

This 8-bit register sets the minimum time gap that is to be allowed between the start of the last transaction and the EOF for full speed transactions.

Table 262. Bit Descriptions for FS_EOF1

Bits	Bit Name	Description	Reset	Access
[7:0]	FS_EOF1	Sets for full speed transactions the time before EOF to stop beginning new transactions, in units of 533.3 ns. The default setting corresponds to 63.46 µs.	0x77	R/W

Software Reset Register

Address: 0x400A007F, Reset: 0x00, Name: SOFT_RST

This 8-bit register asserts the output reset signals NRSTO (reset USB controller logics) and NRSTOX (USB PHY logics). This register is self clearing.

Table 263. Bit Descriptions for SOFT_RST

Bits	Bit Name	Description	Reset	Access
[7:2]	RESERVED	Reserved.	0x0	R/W
1	RSTX	Reset USB PHY logics. This bit resets USB PHY logic in the USBPHYCLK domain. This bit is self clearing.	0x0	R/W1A
0	RST	Reset USB controller logics. This bit resets USB logic in the USBCTLCLK domain. This bit is self clearing.	0x0	R/W1A

Control Status Register for Endpoint 0

Address: 0x400A0102, Reset: 0x0000, Name: CSR0

CSR0 is a 16-bit register that provides control and status bits for Endpoint 0.

Table 264. Bit Descriptions for CSR0

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R/W
8	FLUSHFIFO	Flush FIFO. The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit (Bit 1 and Bit 0) is cleared. Note that FLUSHFIFO must only be used when TXPKTRDY/RXPKTRDY is set. At other times, it may cause data to be corrupted.	0x0	R/W1A
7	SSETUPEND	Serviced setup end. The CPU writes a 1 to this bit to clear the SETUPEND bit. It is cleared automatically.	0x0	R/W1A
6	SRXPKTRDY	Serviced RXPKTRDY. The CPU writes a 1 to this bit to clear the RXPKTRDY bit. It is cleared automatically.	0x0	R/W1A
5	SENDSTALL	Send stall. The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshake is transmitted and then this bit is cleared automatically.	0x0	R/W1A
4	SETUPEND	Setup end. This bit is set when a control transaction ends before the DATAEND bit has been set. An interrupt is generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a 1 to the SSETUPEND bit.	0x0	R/NW
3	DATAEND	Data end. The CPU sets this bit when setting TXPKTRDY for the last data packet, when clearing RXPKTRDY after unloading the last data packet, and when setting TXPKTRDY for a zero length data packet. It is cleared automatically.	0x0	R/W1A
2	SENTSTALL	Sent stall. This bit is set when a STALL handshake is transmitted. The CPU clears this bit.	0x0	R/WC
1	TXPKTRDY	Tx packet ready. The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).	0x0	R/W1A
0	RXPKTRDY	Rx packet ready. This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The CPU clears this bit by setting the SRXPKTRDY bit.	0x0	R/NW

Number of Received Bytes for Endpoint 0 Register

Address: 0x400A0108, Reset: 0x0000, Name: CNT0

CNT0 is a 7-bit read only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RXPKTRDY (CSR0.D0) is set.

Table 265. Bit Descriptions for CNT0

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	R/W
[6:0]	CNT0	Rx Count 0.	0x0	R/NW

Configuration Information for Endpoint 0 Register

Address: 0x400A010F, Reset: 0x02, Name: CFGDATA

CFGDATA is an 8-bit Read-Only register that returns information about the selected core configuration with the Index register set to 0.

Table 266. Bit Descriptions for CFGDATA

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	R0/NW
5	BIGEND	Big endian configuration. Always 0. Indicates little endian ordering. 1: big endian configuration.	0x0	R0/NW
[4:2]	RESERVED	Reserved.	0x0	R1/NW
1	SOFTCON	Soft connect enable. D1 SOFTCON always 1 indicates soft connect/disconnect enabled. 1: soft connect is enabled.	0x1	R1/NW
0	UTMIWID	UTMI data width. Always 0 indicates 8-bit UTMI data width. 0: 8-bit UTMI data width. 1: 16-bit UTMI data width.	0x0	R0/W

Maximum Packet Size for Tx Endpoint 1 Register

Address: 0x400A0110, Reset: 0x0000, Name: TXMAXP1

The TXMAXP1 register defines the maximum amount of data that can be transferred through the selected Tx endpoint in a single operation. There is a TXMAXP register for each Tx Endpoint 1 to Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Tx endpoint, and must not exceed half the FIFO size if double buffering is required.

If this register is changed after packets have been sent from the endpoint, the Tx endpoint FIFO must be completely flushed (using the FLUSHFIFO bit in TXCSR1) after writing the new value to this register.

Note that TXMAXP1 must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 267. Bit Descriptions for TXMAXP1

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED		0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that may be transferred per transaction.	0x0	R/W

Control Status Register for Tx Endpoint 1

Address: 0x400A0112, Reset: 0x0000, Name: TXCSR1

TXCSR is a 16-bit register that provides control and status bits for the Tx endpoint.

Table 268. Bit Descriptions for TXCSR1

Bits	Bit Name	Description	Reset	Access
15	AUTOSET	If this bit is set, TXPKTRDY is automatically set when the maximum data packet size TXMAXP1 is loaded into the Tx FIFO. TXMAXP1 must be a word (4-byte) multiple. If a packet less than the maximum packet size is loaded, the TXPKTRDY bit must be set manually.	0x0	R/W
14	ISO	In peripheral mode, this bit must be set to enable the Tx endpoint for isochronous transfers. This bit must be clear for bulk or interrupt endpoints.	0x0	R/W
13	RESERVED	Reserved.	0x0	R/W
12	DMAREQEN	Set this bit to enable DMA requests for this Tx endpoint.	0x0	R/W
11	FRCDATATGL	Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an acknowledge was received. This can be used by interrupt Tx endpoints that are used to communicate rate feedback for Isochronous endpoints.	0x0	R/W
10	DMAREQMODE	Set this bit to select DMA Request Mode 1, or clear this bit to select DMA Request Mode 0. This bit must not be cleared the cycle before or the same cycle that DMAREQEN is cleared. In DMA Request Mode 0, the DMA is programmed to load one packet at a time. Processor intervention is required for each packet. DMA mode 1 can be used with bulk endpoints to transmit multiple packets without CPU intervention. Note that this bit must not be cleared either before or in the same cycle as the DMAREQEN bit is cleared.	0x0	R/W
[9:7]	RESERVED	Reserved.	0x0	R/W
6	CLRDATATGL	The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.	0x0	R/W1A
5	SENTSTALL	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. The CPU must clear this bit.	0x0	R/W0C
4	SENDSTALL	In device mode, the CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.	0x0	R/W
3	FLUSHFIFO	The CPU writes a 1 to this bit to flush the latest packet from the endpoint TXFIFO. The FIFO pointer is reset, the TXPKTRDY bit is cleared, and an interrupt is generated. This bit can be set simultaneously with TXPKTRDY to abort the packet that is currently being loaded into the FIFO. FLUSHFIFO must only be used when the TXPKTRDY is set. At other times, it may cause data to be corrupted. Also note that if the FIFO is double buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.	0x0	R/W1A

Bits	Bit Name	Description	Reset	Access
2	URUNERR	This bit must be set if an IN token is received when TXPKTRDY is not set. The CPU must clear this bit.	0x0	R/W0C
1	NEFIFO	This bit is set when there is at least 1 packet in the Tx FIFO.	0x0	R/NW
0	TXPKTRDY	The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. If enabled, an interrupt is also generated. TXPKTRDY is also automatically cleared prior to loading a second packet into a double buffered FIFO.	0x0	R/W1S

Maximum Packet Size for Rx Endpoint 1 Register

Address: 0x400A0114, Reset: 0x0000, Name: RXMAXP1

The RXMAXP1 register defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is a RXMAXP1 register for each Rx Endpoint 1 to Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Rx endpoint, and must not exceed half the FIFO size if double buffering is required.

Note that RXMAXP1 must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 269. Bit Descriptions for RXMAXP1

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that can be transferred per transaction.	0x0	R/W

Control Status Register for Rx Endpoint 1

Address: 0x400A0116, Reset: 0x0000, Name: RXCSR1

RXCSR1 is a 16-bit register that provides control and status bits for transfers through Rx endpoint.

Table 270. Bit Descriptions for RXCSR1

Bits	Bit Name	Description			Reset	Access
15	AUTOCLR	If the CPU sets this bit, then the RXPKTRDY bit is automatically cleared when a packet of RXMAXP1 bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY must be cleared automatically. When using the DMA to unload the Rx FIFO, data is read from the Rx FIFO in 4-byte chunks regardless of RXMAXP. Therefore, RXPKTRDY is cleared as follows:			0x0	R/W
		(RXMAXP1%4)	**Actual Bytes Read**	**Packet Sizes that Clear RXPKTRDY**		
		0	RXMAXP1	RXMAXP1, RXMAXP1 − 1, RXMAXP1 − 2, RXMAXP1 − 3		
		3	RXMAXP1 + 1	RXMAXP1, RXMAXP1 − 1, RXMAXP1 − 2		
		2	RXMAXP1 + 2	RXMAXP1, RXMAXP1 − 1		
		1	RXMAXP1 + 3	RXMAXP1		
14	ISO	In peripheral mode, this bit must be set for isochronous transfers and cleared for bulk or interrupt transfers.			0x0	R/W
13	DMAREQEN	The CPU must set this bit to enable DMA requests for the RX endpoint.			0x0	R/W
12	PIDERR	For isochronous transactions, this bit indicates a PID error in the received packet.			0x0	R/W
11	DMAREQMODE	The CPU must set this bit to select DMA Request Mode 1 or clear this bit to select DMA Mode 0. In DMA Mode 0, CPU intervention is required for every packet that is received. For bulk transactions, DMA Mode 1 can be used to transfer multiple RXMAXP1 packets.			0x0	R/W
[10:8]	RESERVED	Reserved.			0x0	R/W
7	CLRDATATGL	If this bit is set, the data toggle bit is cleared.			0x0	R/W1A
6	SENTSTALL	This bit is set when a STALL handshake is transmitted. The CPU must clear this bit.			0x0	R/W0C
5	SENDSTALL	In peripheral mode, this bit must be set to send a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.			0x0	R/W
4	FLUSHFIFO	The CPU must write a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO pointer is cleared and the RXPKTRDY bit is cleared. FLUSHFIFO must only be used when RXPKTRDY is set. At other times, it may cause data to be corrupted. If the FIFO is double buffered, this bit may need to be set twice to completely clear the FIFO.			0x0	R/W1A
3	DATAERR	When in ISO mode, this bit is set when RXPKTRDY is set if the data packet has a CRC or bit stuff error. It is cleared when RXPKTRDY is cleared. This bit is always zero for bulk endpoints.			0x0	R/NW
2	ORUNERR	Overrun error. In peripheral mode, this bit is set if an OUT packet cannot be loaded into the Rx FIFO. The CPU must clear this bit. This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, this bit always returns zero.			0x0	R/W0C
1	FIFOFULL	This bit is set when no more packets can be loaded into the Rx FIFO.			0x0	R/NW
0	RXPKTRDY	This bit is set when a data packet is received. The CPU must clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when this bit is set.			0x0	R/W0C

Number of Byte Received for Rx Endpoint 1 Register

Address: 0x400A0118, Reset: 0x0000, Name: RXCNT1

RXCNT1 is a 14-bit read only register that holds the number of data bytes in the packet currently in line to be read from the Rx FIFO.

Note that the value returned changes as the FIFO is unloaded and is only valid while RXPKTRDY (RxCSR.D0) is set.

Table 271. Bit Descriptions for RXCNT1

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R/W
[13:0]	EPRXCNT	Rx count. For Endpoint 1 to Endpoint 3, this field is 14 bits. For Endpoint 0, this field is seven bits (Bits[6:0]) because the maximum control packet is 64 bytes.	0x0	R/NW

Tx/Rx FIFO Size for Endpoint 1 Register

Address: 0x400A011F, Reset: 0xXX, Name: FIFOSIZE1

FIFOSIZE1 is an 8-bit read only register that returns the sizes of the FIFOs associated with the selected additional Tx/Rx endpoints. The lower nibble encodes the size of the selected Tx endpoint FIFO; the upper nibble encodes the size of the selected Rx endpoint FIFO. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).

- Endpoint 1: Tx 7 (128-byte), Rx 7 (128-byte).
- Endpoint 2: Tx 8 (256-byte), Rx 8 (256-byte).
- Endpoint 3: Tx 9 (512-byte), Rx 9 (512-byte).

Table 272. Bit Descriptions for FIFOSIZE1

Bits	Bit Name	Description	Reset	Access
[7:4]	RXFIFOSZ	Rx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).	0xx	R/NW
[3:0]	TXFIFOSZ	Tx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).	0xx	R/NW

Maximum Packet Size for Tx Endpoint 2 Register

Address: 0x400A0120, Reset: 0x0000, Name: TXMAXP2

The TXMAXP2 register defines the maximum amount of data that can be transferred through the selected TX endpoint in a single operation. There is a TXMAXP2 register for each Tx Endpoint 1 to Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Tx endpoint, and must not exceed half of the FIFO size if double buffering is required.

If this register is changed after packets have been sent from the endpoint, the Tx endpoint FIFO must be completely flushed (using the FLUSHFIFO bit in TXCSR2) after writing the new value to this register.

Note that TXMAXP2 must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 273. Bit Descriptions for TXMAXP2

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that can be transferred per transaction.	0x0	R/W

Control Status Register for Tx Endpoint 2

Address: 0x400A0122, Reset: 0x0000, Name: TXCSR2

TXCSR2 is a 16-bit register that provides control and status bits for the Tx endpoint.

Table 274. Bit Descriptions for TXCSR2

Bits	Bit Name	Description	Reset	Access
15	AUTOSET	If this bit is set, TXPKTRDY is automatically set when the maximum data packet size TXMAXP is loaded into the Tx FIFO. TXMAXP2 must be a word (4-byte) multiple. If a packet less than the maximum packet size is loaded, the TXPKTRDY bit must be set manually.	0x0	R/W
14	ISO	In peripheral mode, this bit must be set to enable the Tx endpoint for isochronous transfers. This bit must be clear for bulk or interrupt endpoints.	0x0	R/W
13	RESERVED	Reserved.	0x0	R/W
12	DMAREQEN	Set this bit to enable DMA requests for this Tx endpoint.	0x0	R/W
11	FRCDATATGL	Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an acknowledge was received. This can be used by interrupt Tx endpoints that are used to communicate rate feedback for isochronous endpoints.	0x0	R/W
10	DMAREQMODE	Set this bit to select DMA Request Mode 1, or clear this bit to select DMA Request Mode 0. This bit must not be cleared the cycle before or the same cycle that DMAREQEN is cleared. In DMA Request Mode 0, the DMA is programmed to load one packet at a time. Processor intervention is required for each packet. DMA Request Mode 1 can be used with bulk endpoints to transmit multiple packets without CPU intervention. Note that this bit must not be cleared either before or in the same cycle as the DMAREQEN bit is cleared.	0x0	R/W
[9:7]	RESERVED	Reserved.	0x0	R/W
6	CLRDATATGL	The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.	0x0	R/W1A
5	SENTSTALL	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. The CPU must clear this bit.	0x0	R/W0C
4	SENDSTALL	In device mode, the CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.	0x0	R/W
3	FLUSHFIFO	The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer is reset, the TXPKTRDY bit is cleared, and an interrupt is generated. This bit may be set simultaneously with TXPKTRDY to abort the packet that is currently being loaded into the FIFO. FLUSHFIFO must only be used when the TXPKTRDY is set. At other times, it may cause data to be corrupted. Also note that if the FIFO is double buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.	0x0	R/W1A
2	URUNERR	This bit is set if an IN token is received when TXPKTRDY is not set. The CPU must clear this bit.	0x0	R/W0C
1	NEFIFO	This bit is set when there is at least 1 packet in the Tx FIFO.	0x0	R/NW
0	TXPKTRDY	The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. If enabled, an interrupt is also generated. TXPKTRDY is also automatically cleared prior to loading a second packet into a double buffered FIFO.	0x0	R/W1S

Maximum Packet Size for Rx Endpoint 2 Register

Address: 0x400A0124, Reset: 0x0000, Name: RXMAXP2

The RXMAXP2 register defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is an RXMAXP2 register for each Rx Endpoint 1 to Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Rx endpoint, and must not exceed half the FIFO size if double buffering is required.

Note that RXMAXP2 must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 275. Bit Descriptions for RXMAXP2

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that can be transferred per transaction.	0x0	R/W

Control Status Register for Rx Endpoint 2

Address: 0x400A0126, Reset: 0x0000, Name: RXCSR2

RXCSR2 is a 16-bit register that provides control and status bits for transfers through Rx endpoint.

Table 276. Bit Descriptions for RXCSR2

Bits	Bit Name	Description	Reset	Access
15	AUTOCLR	If the CPU sets this bit, then the RXPKTRDY bit is automatically cleared when a packet of RXMAXP2 bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY must be cleared automatically. When using the DMA to unload the Rx FIFO, data is read from the Rx FIFO in 4-byte chunks regardless of RXMAXP. Therefore, the RXPKTRDY bit is cleared as follows:	0x0	R/W
		RXMAXP2%4 / **Actual Bytes Read** / **Packet Sizes that Clear RXPKTRDY**		
		0 / RXMAXP2 / RXMAXP2, RXMAXP2 – 1, RXMAXP2 – 2, RXMAXP2 – 3		
		3 / RXMAXP2 + 1 / RXMAXP2, RXMAXP2 – 1, RXMAXP2 – 2		
		2 / RXMAXP2 + 2 / RXMAXP2, RXMAXP2 – 1		
		1 / RXMAXP2 + 3 / RXMAXP2		
14	ISO	In peripheral mode, this bit must be set for isochronous transfers and cleared for bulk or interrupt transfers.	0x0	R/W
13	DMAREQEN	The CPU must set this bit to enable DMA requests for the Rx endpoint.	0x0	R/W
12	PIDERR	For isochronous transactions, this bit indicates a PID error in the received packet.	0x0	R/W
11	DMAREQMODE	The CPU must set this bit to select DMA Request Mode 1 or clear this bit to select DMA Request Mode 0. In DMA Request Mode 0, CPU intervention is required for every packet that is received. For bulk transactions, DMA Request Mode 1 can be used to transfer multiple RXMAXP packets.	0x0	R/W
[10:8]	RESERVED		0x0	R/W
7	CLRDATATGL	If this bit is set, the data toggle bit is cleared.	0x0	R/W1A
6	SENTSTALL	This bit is set when a STALL handshake is transmitted. The CPU must clear this bit.	0x0	R/W0C
5	SENDSTALL	In peripheral mode, this bit must be set to send a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.	0x0	R/W
4	FLUSHFIFO	The CPU must write a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO pointer is cleared and the RXPKTRDY bit is cleared. FLUSHFIFO must only be used when RXPKTRDY is set. At other times, it can cause data to be corrupted. If the FIFO is double buffered, this bit may need to be set twice to completely clear the FIFO.	0x0	R/W1A
3	DATAERR	When in ISO mode, this bit is set when RXPKTRDY is set if the data packet has a CRC or bit stuff error. It is cleared when RXPKTRDY is cleared. This bit is always zero for bulk endpoints.	0x0	R/NW

Bits	Bit Name	Description	Reset	Access
2	ORUNERR	Overrun error. In peripheral mode, this bit is set if an OUT packet cannot be loaded into the Rx FIFO. The CPU must clear this bit. This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, this bit always returns zero.	0x0	R/W0C
1	FIFOFULL	This bit is set when no more packets can be loaded into the Rx FIFO.	0x0	R/NW
0	RXPKTRDY	This bit is set when a data packet is received. The CPU must clear this bit when the packet is unloaded from the Rx FIFO. An interrupt is generated when this bit is set.	0x0	R/W0C

Number of Byte Received for Rx Endpoint 2 Register

Address: 0x400A0128, Reset: 0x0000, Name: RXCNT2

RXCNT2 is a 14-bit read only register that holds the number of data bytes in the packet currently in line to be read from the Rx FIFO.

Note that the value returned changes as the FIFO is unloaded and is only valid while RXPKTRDY (RxCSR.D0) is set.

Table 277. Bit Descriptions for RXCNT2

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R/W
[13:0]	EPRXCNT	Rx count. For Endpoint 1 to Endpoint 3, this field is 14 bits. For Endpoint 0, this field is seven bits (Bits[6:0]) because the maximum control packet is 64 bytes.	0x0	R/NW

Tx/Rx FIFO Size for Endpoint 2 Register

Address: 0x400A012F, Reset: 0xXX, Name: FIFOSIZE2

FIFOSIZE2 is an 8-bit read only register that returns the sizes of the FIFOs associated with the selected additional Tx/Rx endpoints. The lower nibble encodes the size of the selected Tx endpoint FIFO; the upper nibble encodes the size of the selected Rx endpoint FIFO. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).

- Endpoint 1: Tx 7 (128-byte), Rx 7 (128-byte).
- Endpoint 2: Tx 8 (256-byte), Rx 8 (256-byte).
- Endpoint 3: Tx 9 (512-byte), Rx 9 (512-byte).

Table 278. Bit Descriptions for FIFOSIZE2

Bits	Bit Name	Description	Reset	Access
[7:4]	RXFIFOSZ	Rx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).	0xx	R/NW
[3:0]	TXFIFOSZ	Tx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).	0xx	R/NW

Maximum Packet Size for Tx Endpoint 3 Register

Address: 0x400A0130, Reset: 0x0000, Name: TXMAXP3

The TXMAXP3 register defines the maximum amount of data that can be transferred through the selected Tx endpoint in a single operation. There is a TXMAXP3 register for each Tx Endpoint 1 to Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Tx endpoint, and must not exceed half the FIFO size if double buffering is required.

If this register is changed after packets have been sent from the endpoint, the Tx endpoint FIFO must be completely flushed (using the FLUSHFIFO bit in TXCSR3) after writing the new value to this register.

Note that TXMAXP3 must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 279. Bit Descriptions for TXMAXP3

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that can be transferred per transaction.	0x0	R/W

Control Status Register for Tx Endpoint 3

Address: 0x400A0132, Reset: 0x0000, Name: TXCSR3

TXCSR3 is a 16-bit register that provides control and status bits for the Tx endpoint.

Table 280. Bit Descriptions for TXCSR3

Bits	Bit Name	Description	Reset	Access
15	AUTOSET	If this bit is set, TXPKTRDY is automatically set when the maximum data packet size TXMAXP3 is loaded into the Tx FIFO. TXMAXP3 must be a word (4-byte) multiple. If a packet less than the maximum packet size is loaded, the TXPKTRDY bit must be set manually.	0x0	R/W
14	ISO	In peripheral mode, this bit must be set to enable the Tx endpoint for isochronous transfers. This bit must be clear for bulk or interrupt endpoints.	0x0	R/W
13	RESERVED	Reserved.	0x0	R/W
12	DMAREQEN	Set this bit to enable DMA requests for this Tx endpoint.	0x0	R/W
11	FRCDATATGL	Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an acknowledge was received. This can be used by interrupt Tx endpoints that are used to communicate rate feedback for isochronous endpoints.	0x0	R/W
10	DMAREQMODE	Set this bit to select DMA Request Mode 1, or clear this bit to select DMA Request Mode 0. This bit must not be cleared the cycle before or the same cycle that DMAREQEN is cleared. In DMA Request Mode 0, the DMA is programmed to load one packet at a time. Processor intervention is required for each packet. DMA Request Mode 1 can be used with bulk endpoints to transmit multiple packets without CPU intervention. Note that this bit must not be cleared either before or in the same cycle as the above the DMAREQEN bit is cleared.	0x0	R/W
[9:7]	RESERVED	Reserved.	0x0	R/W
6	CLRDATATGL	The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.	0x0	R/W1A
5	SENTSTALL	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. The CPU must clear this bit.	0x0	R/W0C
4	SENDSTALL	In device mode, the CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.	0x0	R/W

Bits	Bit Name	Description	Reset	Access
3	FLUSHFIFO	The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer is reset, the TXPKTRDY bit is cleared, and an interrupt is generated. This bit can be set simultaneously with TXPKTRDY to abort the packet that is currently being loaded into the FIFO. FLUSHFIFO must only be used when TXPKTRDY is set. At other times, it can cause data to be corrupted. Also note that if the FIFO is double buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.	0x0	R/W1A
2	URUNERR	This bit is set if an IN token is received when TXPKTRDY is not set. The CPU must clear this bit.	0x0	R/W0C
1	NEFIFO	This bit is set when there is at least one packet in the Tx FIFO.	0x0	R/NW
0	TXPKTRDY	The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. If enabled, an interrupt is also generated. TXPKTRDY is also automatically cleared prior to loading a second packet into a double buffered FIFO.	0x0	R/W1S

Maximum Packet Size for Rx Endpoint 3 Register

Address: 0x400A0134, Reset: 0x0000, Name: RXMAXP3

The RXMAXP3 register defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is an RXMAXP3 register for each Rx Endpoint 1 to Endpoint 3 (except Endpoint 0).

Bits[10:0] define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt, and isochronous transfers in full speed operations.

The value written to Bits[10:0] must match the value given in the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch can cause unexpected results.

The total amount of data represented by the value written to this register must not exceed the FIFO size for the Rx endpoint, and must not exceed half the FIFO size if double buffering is required.

Note that RXMAXP3 must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

Table 281. Bit Descriptions for RXMAXP3

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:0]	MAXPAYLOAD	Maximum number of bytes that can be transferred per transaction.	0x0	R/W

Control Status Register for Rx Endpoint 3

Address: 0x400A0136, Reset: 0x0000, Name: RXCSR3

RXCSR3 is a 16-bit register that provides control and status bits for transfers through Rx endpoint.

Table 282. Bit Descriptions for RXCSR3

Bits	Bit Name	Description			Reset	Access
15	AUTOCLR	If the CPU sets this bit, then the RXPKTRDY bit is automatically cleared when a packet of RXMAXP3 bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY must be cleared automatically. When using the DMA to unload the Rx FIFO, data is read from the Rx FIFO in 4 byte chunks regardless of RXMAXP3. Therefore, the RXPKTRDY bit is cleared as follows:			0x0	R/W
		RXMAXP3%4	**Actual Bytes Read**	**Packet Sizes that Clear RXPKTRDY**		
		0	RXMAXP3	RXMAXP3, RXMAXP3 – 1, RXMAXP3 – 2, RXMAXP3 – 3		
		3	RXMAXP3 + 1	RXMAXP3, RXMAXP3 – 1, RXMAXP3 – 2		
		2	RXMAXP3 + 2	RXMAXP3, RXMAXP3 – 1		
		1	RXMAXP3 + 3	RXMAXP3		
14	ISO	In peripheral mode, this bit must be set for isochronous transfers and cleared for bulk or interrupt transfers.			0x0	R/W
13	DMAREQEN	The CPU must set this bit to enable DMA requests for the Rx endpoint.			0x0	R/W
12	PIDERR	For isochronous transactions, this bit indicates a PID error in the received packet.			0x0	R/W

Bits	Bit Name	Description	Reset	Access
11	DMAREQMODE	The CPU must set this bit to select DMA Request Mode 1 or clear this bit to select DMA Request Mode 0. In DMA Request Mode 0, CPU intervention is required for every packet that is received. For bulk transactions, DMA Request Mode 1 can be used to transfer multiple RXMAXP3 packets.	0x0	R/W
[10:8]	RESERVED	Reserved.	0x0	R/W
7	CLRDATATGL	If this bit is set, the data toggle bit is cleared.	0x0	R/W1A
6	SENTSTALL	This bit is set when a STALL handshake is transmitted. The CPU must clear this bit.	0x0	R/W0C
5	SENDSTALL	In peripheral mode, this bit must be set to send a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect for isochronous transfers.	0x0	R/W
4	FLUSHFIFO	The CPU must write a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO pointer is cleared and the RXPKTRDY bit is cleared. FLUSHFIFO must only be used when RXPKTRDY is set. At other times, it can cause data to be corrupted. If the FIFO is double buffered, this bit may need to be set twice to completely clear the FIFO.	0x0	R/W1A
3	DATAERR	When in ISO mode, this bit is set when RXPKTRDY is set if the data packet has a CRC or bit stuff error. It is cleared when RXPKTRDY is cleared. This bit is always zero for bulk endpoints.	0x0	R/NW
2	ORUNERR	Overrun error. In peripheral mode, this bit is set if an OUT packet cannot be loaded into the Rx FIFO. The CPU must clear this bit. This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, this bit always returns zero.	0x0	R/W0C
1	FIFOFULL	This bit is set when no more packets can be loaded into the Rx FIFO.	0x0	R/NW
0	RXPKTRDY	This bit is set when a data packet is received. The CPU must clear this bit when the packet is unloaded from the Rx FIFO. An interrupt is generated when this bit is set.	0x0	R/W0C

Number of Byte Received for Rx Endpoint 3 Register

Address: 0x400A0138, Reset: 0x0000, Name: RXCNT3

RXCNT3 is a 14-bit read only register that holds the number of data bytes in the packet currently in line to be read from the Rx FIFO.

Note that the value returned changes as the FIFO is unloaded and is only valid while RXPKTRDY (RxCSR.D0) is set.

Table 283. Bit Descriptions for RXCNT3

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R/W
[13:0]	EPRXCNT	Rx count. For Endpoint 1 to Endpoint 3, this field is 14 bits. For Endpoint 0, this field is seven bits (Bits[6:0]) because the maximum control packet is 64 bytes.	0x0	R/NW

Tx/Rx FIFO Size for Endpoint 3 Register

Address: 0x400A013F, Reset: 0xXX, Name: FIFOSIZE3

FIFOSIZE3 is an 8-bit read only register that returns the sizes of the FIFOs associated with the selected additional Tx/Rx endpoints. The lower nibble encodes the size of the selected Tx endpoint FIFO; the upper nibble encodes the size of the selected Rx endpoint FIFO. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).

- Endpoint 1: Tx 7 (128-byte), Rx 7 (128-byte).
- Endpoint 2: Tx 8 (256-byte), Rx 8 (256-byte).
- Endpoint 3: Tx 9 (512-byte), Rx 9 (512-byte).

Table 284. Bit Descriptions for FIFOSIZE3

Bits	Bit Name	Description	Reset	Access
[7:4]	RXFIFOSZ	Rx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).	0xx	R/NW
[3:0]	TXFIFOSZ	Tx FIFO size. Values of 3 to 13 correspond to a FIFO size of 2^n bytes (8 bytes to 8192 bytes).	0xx	R/NW

DMA Interrupt Register

Address: 0x400A0200, Reset: 0x00, Name: DMA_IRQ

This register provides an interrupt for each DMA channel. This interrupt register is cleared when read. When any bit of this register is set, the USB DMA INTERRUPT is asserted. Bits in this register are only set if the DMA interrupt enable bit for the corresponding channel is enabled (Register DMA_CNTL.D3).

Table 285. Bit Descriptions for DMA_IRQ

Bits	Bit Name	Description	Reset	Access
[7:2]	RESERVED	Reserved.	0x0	R/W
1	D1	Channel 1 DMA interrupt.	0x0	RC/NW
0	D0	Channel 0 DMA interrupt.	0x0	RC/NW

DMA Control for Channel 0 Register

Address: 0x400A0204, Reset: 0x0000, Name: DMA_CTL0

DMA transfer control for transfer direction, transfer mode, DMA burst modes.

Table 286. Bit Descriptions for DMA_CTL0

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED		0x0	R/W
[10:9]	BRSTM	Burst mode. Burst mode, sets allowable DMA_HBURST[2:0] (AHB bus signal) the DMA controller uses. 00: bursts of unspecified length. 01: INCR4 or unspecified length. 10: INCR8, INCR4, or unspecified length. 11: INCR16, INCR8, INCR4, or unspecified length.	0x0	R/W
8	ERR	Bus error. Indicates an error has been observed on the DMA_HRESPM[1:0] (AHB bus signal). This bit is cleared by software. 0: no error. 1: error occurred.	0x0	R/W
[7:4]	EP	DMA endpoint assignment. The endpoint this DMA channel is assigned to.	0x0	R/W
3	IE	DMA interrupt enable. 0: disable. 1: enable.	0x0	R/W
2	MODE	DMA mode. This bit selects the DMA transfer mode. 0: DMA Mode 0, the DMA only loads/unloads one packet. 1: In DMA Mode 1, the DMA controller can load/unload an entire bulk transfer, which can be multiple packets at size of MAXP. DMA Mode 1 can only be used with bulk endpoints.	0x0	R/W
1	DIR	DMA transfer direction. This bit selects the DMA transfer direction. 0: DMA write = Rx endpoint. 1: DMA read = Tx endpoint.	0x0	R/W
0	EN	DMA enable. This bit enables the DMA and causes the transfer to begin. 0: disable. 1: enable.	0x0	R/W

DMA Address for Channel 0 Register

Address: 0x400A0208, Reset: 0x00000000, Name: DMA_ADDR0

The address of the block of memory to read data from or write data to. The address must be aligned to 32-bit words, the lower two address bits are read only. This register increments as the DMA transfer progresses.

Table 287. Bit Descriptions for DMA_ADDR0

Bits	Bit Name	Description	Reset	Access
[31:0]	DMA_ADDR	DMA source or destination starting address. Note that Bits[1:0] of DMA_ADDR are read only.	0x0	R/W

DMA Count for Channel 0 Register

Address: 0x400A020C, Reset: 0x00000000, Name: DMA_CNT0

This register identifies the current DMA count of the transfer. Software sets the initial count of the transfer, which identifies the entire transfer length. As the count progresses, this count is decremented as bytes are transferred.

If this field is set to zero, no data is transferred and an USB DMA interrupt is generated.

Table 288. Bit Descriptions for DMA_CNT0

Bits	Bit Name	Description	Reset	Access
[31:0]	DMA_COUNT	DMA data count.	0x0	R/W

DMA Control for Channel 1 Register

Address: 0x400A0214, Reset: 0x0000, Name: DMA_CTL1

DMA transfer control for transfer direction, transfer mode, DMA burst modes.

Table 289. Bit Descriptions for DMA_CTL1

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R/W
[10:9]	BRSTM	Burst mode. Burst mode sets allowable DMA_HBURST[2:0] the DMA controller uses. 00: bursts of unspecified length. 01: INCR4 or unspecified length. 10: INCR8, INCR4, or unspecified length. 11: INCR16, INCR8, INCR4, or unspecified length.	0x0	R/W
8	ERR	Bus Error. Bus error indicates an error has been observed on the DMA_HRESPM[1:0]. This bit is cleared by software. 0: no error. 1: error occurred.	0x0	R/W
[7:4]	EP	DMA endpoint assignment. The endpoint this DMA channel is assigned to.	0x0	R/W
3	IE	DMA interrupt enable. 0: disable. 1: enable.	0x0	R/W
2	MODE	DMA mode. This bit selects the DMA transfer mode. 0: in DMA Mode 0, the DMA only loads/unloads one packet. 1: in DMA Mode 1, the DMA controller can load/unload an entire bulk transfer, which can be multiple packets at size of MAXP. DMA Mode 1 can only be used with bulk endpoints.	0x0	R/W
1	DIR	DMA transfer direction. This bit selects the DMA transfer direction. 0: DMA write = Rx endpoint. 1: DMA read = Tx endpoint.	0x0	R/W
0	EN	DMA enable. This bit enables the DMA and causes the transfer to begin. 0: disable. 1: enable.	0x0	R/W

DMA Address for Channel 1 Register

Address: 0x400A0218, Reset: 0x00000000, Name: DMA_ADDR1

The address of the block of memory to read data from or write data to. The address must be aligned to 32-bit words, the lower two address bits are read only. This register increments as the DMA transfer progresses.

Table 290. Bit Descriptions for DMA_ADDR1

Bits	Bit Name	Description	Reset	Access
[31:0]	DMA_ADDR	DMA source or destination starting address. Note that Bits[1:0] of DMA_ADDR are read only.	0x0	R/W

DMA Count for Channel 1 Register

Address: 0x400A021C, Reset: 0x00000000, Name: DMA_CNT1

This register identifies the current DMA count of the transfer. Software sets the initial count of the transfer, which identifies the entire transfer length. As the count progresses, this count is decremented as bytes are transferred.

If this field is set to zero, no data is transferred and a USB DMA interrupt is generated.

Table 291. Bit Descriptions for DMA_CNT1

Bits	Bit Name	Description	Reset	Access
[31:0]	DMA_COUNT	DMA data count.	0x0	R/W

Rx Double Packet Buffer Disable for Endpoints 1 to 3 Register

Address: 0x400A0340, Reset: 0x0000, Name: RXDPKTBUFDIS

When asserted (RXDPKTBUFDIS = 1), the register disables double packet buffering for the corresponding endpoint regardless of the endpoint FIFO size and the RX MAXP size relationship. When deasserted (RXDPKTBUFDIS = 0), this register does not necessarily enable double packet buffering, but rather allows double packet buffering to be determined based upon the endpoint FIFO size and RX MAXP size relationship.

Table 292. Bit Descriptions for RXDPKTBUFDIS

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R/W
3	EP3RXDBDIS	Disable Rx double buffer of Endpoint 3. 0: double buffer enabled. 1: double buffer disabled.	0x0	R/W
2	EP2RXDBDIS	Disable Rx double buffer of Endpoint 2. 0: double buffer enabled. 1: double buffer disabled.	0x0	R/W
1	EP1RXDBDIS	Disable Rx double buffer of Endpoint 1. 0: double buffer enabled. 1: double buffer disabled.	0x0	R/W
0	RESERVED	Reserved.	0x0	R/W

Tx Double Packet Buffer Disable for Endpoints 1 to 3 Register

Address: 0x400A0342, Reset: 0x0000, Name: TXDPKTBUFDIS

When asserted (TXDPKTBUFDIS = 1), the register disables double packet buffering for the corresponding endpoint regardless of the endpoint FIFO size and the TX MAXP size relationship. When deasserted (TXDPKTBUFDIS = 0), this register does not necessarily enable double packet buffering, but rather allows double packet buffering to be determined based upon the endpoint FIFO size and TX MAXP size relationship.

Table 293. Bit Descriptions for TXDPKTBUFDIS

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R/W
3	EP3TXDBDIS	Disable Tx double buffer of Endpoint 3. 0: double buffer enabled. 1: double buffer disabled.	0x0	R/W
2	EP2TXDBDIS	Disable Tx double buffer of Endpoint 2. 0: double buffer enabled. 1: double buffer disabled.	0x0	R/W
1	EP1TXDBDIS	Disable Tx double buffer of Endpoint 1. 0: double buffer enabled. 1: double buffer disabled.	0x0	R/W
0	RESERVED	Reserved.	0x0	R/W

Chirp Timeout Register

Address: 0x400A0344, Reset: 0x4074, Name: CT_UCH

This register sets the chirp timeout. This number, when multiplied by 4, represents the number of USBPHYCLK cycles before the timeout occurs. That is, because USBPHYCLK is 60 MHz, this number represents the number of 67 ns time intervals before the timeout occurs.

Table 294. Bit Descriptions for CT_UCH

Bits	Bit Name	Description	Reset	Access
[15:0]	C_T_UCH	Chirp timeout. Configurable chirp timeout timer; The default value is 4074h (USBPHYCLK is 60 MHz) corresponding to a delay of 1.1 ms.	0x4074	R/W

LPM Attribute Register

Address: 0x400A0360, Reset: 0x0000, Name: LPM_ATTR

The attributes of an LPM transaction and sleep cycle. It contains the equivalent attributes that were received in the last LPM transaction that was accepted. This register is updated with the LPM packet contents if the response to the LPM transaction was an acknowledge. This register can be updated via software. In all other cases, this register holds its current value.

Table 295. Bit Descriptions for LPM_ATTR

Bits	Bit Name	Description	Reset	Access
[15:12]	EP	Endpoint. The endpoint in the token packet of the LPM transaction.	0x0	R/W
[11:9]	RESERVED	Reserved.	0x0	R/W
8	RMTWAK	Remote wake-up enable. This bit is the remote wake-up enable bit. RMTWAK = 0 means remote wake-up is not enabled. RMTWAK = 1 means remote wake-up is enabled. This bit is applied on a temporary basis only and is only applied to the current suspend state. After the current suspend cycle, the remote wake-up capability that was negotiated upon enumeration applies.	0x0	R/W
[7:4]	HIRD	Host initiated resume duration. This value is the minimum time the host drives resume on the bus. The value in these bit corresponds to an actual resume time as follows: resume time = 50 µs + HIRD × 75 µs. This results in a range of 50 µs to 1200 µs.	0x0	R/W
[3:0]	LINKSTATE	This value is provided by the host to the peripheral to indicate what state the peripheral must transition to after the receipt and acceptance of an LPM transaction. 0001 indicates sleep state (L1). All other combinations are reserved.	0x0	R/W

LPM Control Register

Address: 0x400A0362, Reset: 0x00, Name: LPM_CTL

Table 296. Bit Descriptions for LPM_CTL

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x0	R/W
4	LPMNAK	This bit is used to place all endpoints in a state such that the response to all transactions other than an LPM transaction are NAK. This bit only takes effect after the USB controller has been LPM suspended. In this case, the USB controller continues to NAK until this bit is cleared by software.	0x0	R/W
[3:2]	LPMEN	These bits are used to enable LPM in the USB controller. There are three levels in which LPM can be enabled that determine the response to LPM transactions. 00: LPM and extended transactions are not supported. In this case, the controller does not respond to LPM transactions and Transaction 01 times out. 01: LPM and extended transactions are not supported. In this case, the controller does not respond to LPM transactions and Transaction 01 times out. 10: LPM is not supported but extended transactions are supported. In this case, the controller responds to an LPM transaction with a STALL. 11: the controller supports LPM extended transactions. In this case, the controller responds with a NYET or an acknowledge as determined by the value of LPMXMT and other conditions.	0x0	R/W
1	LPMRES	This bit is used by software to initiate resume (remote wake-up). This bit differs from the classic RESUME bit in the POWER register in that the RESUME signal timing is controlled by hardware. When software writes this bit, resume signaling is asserted for 50 µs. This bit is self clearing.	0x0	R/W
0	LPMXMT	This bit is set by software to instruct the USB controller to transition to the L1 state upon the receipt of the next LPM transaction. This bit is only effective if LPMEN is set to 11. This bit can be set in the same cycle as LPMEN. If this bit is set to 1 and LPMEN = 11, the USB controller can respond in the following ways. If no data is pending (all Tx FIFOs are empty), the USB controller responds with an acknowledge. In this case, this bit self clears and a software interrupt is generated. If data is pending (data resides in at least one Tx FIFO), the USB controller responds with a NYET. In this case, this bit does not self clear; however, a software interrupt is generated.	0x0	R/W

LPM Interrupt Enable Register

Address: 0x400A0363, Reset: 0x00, Name: LPM_IEN

The LPM_IEN is a 6-bit register that provides enable bits for the interrupts in the LPM_IRQ register. If a bit in this register is set to 1, USB INTTERRUPT is asserted when the corresponding interrupt in the LPM_INTR register is set. If a bit in this register is set to 0, the corresponding register in LPM_INTR is still set but USB INTTERRUPT is not be asserted.

Table 297. Bit Descriptions for LPM_IEN

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	R/NW
5	LPMERREN	Enable LPMERR interrupt.	0x0	R/W
4	LPMRESEN	Enable LPMRES interrupt.	0x0	R/W
3	LPMNCEN	Enable LPMNC interrupt.	0x0	R/W
2	LPMACKEN	Enable LPMACK interrupt.	0x0	R/W
1	LPMNYEN	Enable LPMNY interrupt.	0x0	R/W
0	LPMSTEN	Enable LPMST interrupt.	0x0	R/W

LPM Interrupt Register

Address: 0x400A0364, Reset: 0x00, Name: LPM_IRQ

The LPM_IRQ is a 7-bit register that provides status of the LPM power state. When a bit is set to 1, if the corresponding enable bit is also set to 1, the output USB INTERRUPT is asserted (low). If the corresponding enable bit is set to 0, the output USB INTERRUPT is not asserted. This register is clear on read.

Table 298. Bit Descriptions for LPM_IRQ

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	R/W
5	LPMERR	This bit is set if an LPM transaction is received that has a LINKSTATE field that is not supported. In this case, the response to the transaction is a STALL. However, the LPM_ATTR register is updated so that software can observe the noncompliant LPM packet payload.	0x0	RC/NW
4	LPMRES	This bit is set if the controller has been resumed for any reason. This bit is mutually exclusive from the RESUME bit in the POWER register.	0x0	RC/NW
3	LPMNC	This bit is set when an LPM transaction is received and the controller responds with a NYET due to data pending in the Rx FIFOs. This can only occur under the following conditions: the LPMEN field in the LPM_CNTRL register is set to 11, the LPMXMT field is set to 1, and there was data pending in the controller Tx FIFOs.	0x0	RC/NW
2	LPMACK	This bit is set when an LPM transaction is received and the controller responds with an acknowledge. This can only occur under the following conditions: the LPMEN field in the LPM_CNTRL register is set to 11, the LPMXMT field is set to 1, and there was no data pending in the controller Tx FIFOs.	0x0	RC/NW
1	LPMNY	This bit is set when an LPM transaction is received and the controller responds with a NYET. This can only occur under the following conditions: the LPMEN field in the LPM_CNTRL register is set to 11 and the LPMXMT field is set to 0.	0x0	RC/NW
0	LPMST	This bit is set when an LPM transaction is received and the controller responds with a STALL. This can only occur under the following condition: the LPMEN field in the LPM_CNTRL register is set to 01.	0x0	RC/NW

USB PHY Control Register

Address: 0x400A039C, Reset: 0x0000, Name: USBPHYCTL

Table 299. Bit Descriptions for USBPHYCTL

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R/W
7	PU_LO	This bit has no effect if PUCON = 0 or PHY_MAN = 0. 0: higher resistance pull-up. 1: lower resistance pull-up.	0x0	R/W
6	PUCON	Actual pull-up resistance depends on PU_LO. This bit has no effect if PHY_MAN = 0. 0: no pull-up resistor. 1: 1.5 kΩ pull-up resistor on D+.	0x0	R/W
5	PDCON	This bit has no effect if PHY_MAN = 0. 0: no pull-down resistor. 1: 15 kΩ pull-down resistor on D+ (14.25 kΩ to 24.80 kΩ).	0x0	R/W
4	NDOE	When this bit = 0, the differential input comparator is powered down (DIDIF). This bit has no effect if PHY_MAN = 0 or SUSPEND = 1. 0: output driver enable. 1: output driver disable.	0x0	R/W
3	DOM	Level to output onto the D− (DOM) pin. The output driver is only enabled if NDOE = 0 and SUSPEND = 0. This bit has no effect if PHY_MAN = 0. 0: drive D− low. 1: drive D− high.	0x0	R/W
2	DOP	Level to output onto the USB D+ (DOP) pin. The output driver is only enabled if NDOE = 0 and SUSPEND = 0. This bit has no effect if PHY_MAN = 0. 0: drive D+ low. 1: drive D+ high.	0x0	R/W

Bits	Bit Name	Description	Reset	Access
1	SUSPEND	SUSPEND powers down the output drivers, differential input comparator, ID input comparator, and pull-up resistor. This bit has no effect if PHY_MAN = 0. 0: normal operation (power enabled). 1: SUSPEND enabled.	0x0	R/W
0	PHY_MAN	0: PHY inputs come from controller. 1: PHY inputs come from this register.	0x0	R/W

USB PHY Status Register

Address: 0x400A039E, Reset: 0x00XX, Name: USBPHYSTAT

Table 300. Bit Descriptions for USBPHYSTAT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R/NW
7	VBUSVALID	0: V_{BUS} < 3.8 V. 1: V_{BUS} > 3.8 V.	0xx	R/NW
[6:4]	RESERVED	Reserved.	0x0	R/NW
2	DIDIF	Output of differential comparator on D+ and D−. The differential comparator is disabled in suspend and this bit is forced to zero.	0xx	R/NW
1	DIM	Single ended D− (DIM) input. Always enabled even in suspend.	0xx	R/NW
0	DIP	Single ended D+ input. Always enabled even in suspend.	0xx	R/NW

RAM Address Register

Address: 0x400A03B0, Reset: 0x00000000, Name: RAM_ADDR

Table 301. Bit Descriptions for RAM_ADDR

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R/W
[19:17]	CNT	This field specifies the number of accesses to RAMDATA before RAMADDR is automatically incremented/decremented. A read or write to the RAMDATA register is counted as one access. The auto increment/decrement function can be used to skip writing RAMADDR when iterating through memory for marching RAM tests. If this field is set to zero, the auto increment/decrement feature is disabled.	0x0	R/W
16	DIR	If CNT is nonzero, this bit determines the direction for the address automatic increment/decrement. 0: automatic decrement. 1: automatic increment.	0x0	R/W
[15:9]	RESERVED	Reserved.	0x0	R/W
[8:0]	RAMADDR	Ram word address. This address can access 4000 words (16,000 bytes). This register can be automatically incremented or decremented after CNT accesses to the RAM_DATA register.	0x0	R/W

RAM Data Register

Address: 0x400A03B4, Reset: 0x0000000X, Name: RAM_DATA

Table 302. Bit Descriptions for RAM_DATA

Bits	Bit Name	Description	Reset	Access
[31:0]	RAMDATA	Data to write or read from the FIFO RAM at RAMADDR.	0xx	R/W

UART

FEATURES

The UART peripheral is a serial full duplex universal asynchronous receiver/transmitter that is compatible with the industry standard 16450. The serial communication follows an asynchronous protocol, supporting various word length, stop bits, and parity generation options.

This UART also contains interrupt handling hardware and provides a fractional divider that facilitates high accuracy baud rate generation.

Interrupts can be generated from a number of unique events, such as data buffer full/empty, transfer error detection, and break detection.

While the UART implementation supports modem control signals, only serial Tx and Rx functionality is supported. Modem inputs (DCD, RI, DSR, CTS) are tied high internally, and modem outputs (RTS, DTR, OUT1, OUT2) are not connected.

OPERATION

Serial Communications

The asynchronous serial communication protocol supports the following options:

- 5 data bits to 8 data bits.
- 1, 2 or 1 and 1/2 stop bits.
- None, or even or odd parity.
- Baud rate = (UCLK/((2 × (M + N/2048)) × 16 × COMDIV)), where COMDIV = 1 to 65,536; M = 1 to 3; and N = 0 to 2047.

All data-words require a start bit and at least one stop bit. This creates a range from 7 bits to 12 bits for each word.

Transmit operation is initiated by writing to the transmit holding register (COMTX). After a synchronization delay, the data is moved to the transmit shift register, where it is shifted out at a baud (bit) rate equal to UCLK/(2 × 16 × COMDIV)/(M + N/2048) with start, stop, and parity bits appended as required. All data-words begin with a low going start bit. The transfer of the transmit holding register to the transmit shift register causes the transmit register empty status bit (THRE) to be set. Table 303 shows some example baud rates, assuming a 16 MHz input clock.

Receive operation uses the same data format as the transmit configuration, except for the number of stop bits, which is always one. After the start bit is detected, the received word is shifted into the receive shift register. After the appropriate number of bits (including stop bits) are received, the receive shift register is transferred to the receive buffer register. After the appropriate synchronization delay, the receive buffer register full status flag (STA) is updated.

A sampling clock equal to 16 times the baud rate is used to sample the data as close to the midpoint of the bit as possible. A receive filter is also present that removes spurious pulses of less than two times the sampling clock period.

Note that data is transmitted and received least significant bit first, that is, transmit shift register, Bit 0. This is often not the assumed case by the user. This, however, is standard for the protocol.

Baud Rate Generator

To bypass the fractional divider, set Bit 15 of the COMFBR register to 0 (FBEN). This results in a baud rate = (UCLK/(2 × 16 × COMDIV)).

DIVN and DIVM in the fractional baud rate generator can be used with COMDIV = 1 to achieve a baud rate value closer to the desired rate. When Bit 15 of the COMFBR register is 1 (FBEN), typically DIVM = 1, and N/2048 = UCLK/(baud rate × 2 × 16 × COMDIV); alternatively, try using DIVM = 2 and DIVM = 3 to achieve better baud rate accuracy.

Table 303. Baud Rate Examples Based on a 16 MHz UCLK

Baud Rates	COMDIV	DIVM	DIVN	Actual	% Error
9600	17	3	131	9599.25	−0.0078%
19200	8	3	523	19199.04	−0.0050%
38400	4	3	523	38398.08	−0.0050%
57600	8	1	174	57605.76	+0.0100%
115200	2	2	348	115211.5	+0.0100%
230400	2	1	174	230423	+0.0100%
460800	1	1	174	460846.1	+0.0100%

IO Mode

In this mode, the software controls moving data to and from the UART. This is accomplished by interrupt service routines that respond to the transmit and receive interrupts by either reading or writing data as appropriate. This mode puts certain constraints on the software itself, in that the software must respond within a certain time to prevent overrun errors from occurring in the receive channel.

IO mode also requires polling the status flags to determine when it is okay to move data.

This mode is processor intensive and is not typically used unless the system can tolerate the overhead. Interrupts can be disabled using the UART interrupt enable register (COMIEN).

Writing the transmit holding register when it is not empty or reading the receive buffer register when it is not full produces an incorrect result and, therefore, must be avoided. In the former case, the transmit holding register is overwritten by the new word, and the previous word is never transmitted. In the latter case, the previously received word is read again. Both of these errors must be avoided in software by correctly using either interrupts or status register polling. These errors are not detected in hardware.

DMA Mode

In this mode, user code does not move data to and from the UART, rather DMA request signals to the DMA block are generated, indicating that the UART is ready to transmit or receive data. These DMA request signals can be disabled in the UART interrupt enable register (COMIEN).

Interrupts

The UART peripheral has one output signal from the interrupt controller of the core that represents all Rx and Tx interrupts. The UART interrupt identification register (COMIIR) must be read by the software to determine the cause of the interrupt. Note that in DMA mode, the break and modem status interrupts are not available.

In IO mode, interrupts may be generated for the following cases:

- Receive buffer register full.
- Receive overrun error.
- Receive parity error.
- Receive framing error.
- Break interrupt (RXSIN held low).
- Modem status interrupt (changes to DCD, RI, DSR, or CTS).
- Transmit holding register empty.

Buffer Requirements

This UART is double buffered (holding register and shift register).

UART MEMORY MAPPED REGISTERS

UART Register Map

Table 304. UART0 Register Summary

Address	Name	Description	Reset	RW
0x40005000	COMTX	Transmit holding register	0x0000	W
0x40005000	COMRX	Receive buffer register	0x0000	R
0x40005004	COMIEN	Interrupt enable	0x0000	RW
0x40005008	COMIIR	Interrupt ID	0x0001	RC
0x4000500C	COMLCR	Line control	0x0000	RW
0x40005010	COMMCR	Modem control	0x0000	RW
0x40005014	COMLSR	Line status	0x0060	RC
0x40005018	COMMSR	Modem status	0x0000	RC
0x4000501C	COMSCR	Scratch buffer	0x0000	RW
0x40005020	COMMCFG	SOUT modulation configuration	0x0000	RW
0x40005024	COMFBR	Fractional baud rate	0x0000	RW
0x40005028	COMDIV	Baud rate divider	0x0001	RW
0x40005030	COMCTL	UART control register	0x0000	RW

Transmit Holding Register

Address: 0x40005000, Reset: 0x0000, Name: COMTX

Table 305. Bit Descriptions for COMTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	THR	Transmit holding register.	0x0	W

Receive Buffer Register

Address: 0x40005000, Reset: 0x0000, Name: COMRX

Table 306. Bit Descriptions for COMRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	RBR	Receive buffer register.	0x0	R

Interrupt Enable Register

Address: 0x40005004, Reset: 0x0000, Name: COMIEN

Table 307. Bit Descriptions for COMIEN

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved.	0x0	R
5	EDMAR	DMA requests in receive mode. 0: DMA requests enabled. 1: DMA requests disabled.	0x0	RW
4	EDMAT	DMA requests in transmit mode. 0: DMA requests are enabled. 1: DMA requests are disabled.	0x0	RW
3	EDSSI	Modem status interrupt. Interrupt is generated when any of Bits[3:0] of COMMSR are set. 0: interrupt enabled. 1: interrupt disabled.	0x0	RW
2	ELSI	Rx status interrupt. 0: interrupt enabled. 1: interrupt disabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
1	ETBEI	Transmit buffer empty interrupt. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
0	ERBFI	Receive buffer full interrupt. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Interrupt ID Register

Address: 0x40005008, Reset: 0x0001, Name: COMIIR

Table 308. Bit Descriptions for COMIIR

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:1]	STA	Interrupt status. When NIRQ is low (active low), this indicates an interrupt and the STA bit decoding is used. 00: modem status interrupt (Read COMMSR to clear). 01: transmit buffer empty interrupt (Write to COMTX or read COMIIR to clear). 10: receive buffer full interrupt (Read COMRX to clear). 11: receive line status interrupt (Read COMLSR to clear).	0x0	RC
0	NIRQ	Interrupt flag.	0x1	RC

Line Control Register

Address: 0x4000500C, Reset: 0x0000, Name: COMLCR

Table 309. Bit Descriptions for COMLCR

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	R
6	BRK	Set break. 0: force TxD (the UART transmit pin) to 0. 1: normal TxD operation.	0x0	RW
5	SP	Stick parity. Used to force parity to defined values. When set, the parity is based on the following bit settings. If EPS = 1 and PEN = 1, parity is forced to 0. If EPS = 0 and PEN = 1, parity is forced to 1. If EPS = X and PEN = 0, no parity is transmitted. 0: parity is not forced based on EPS and PEN. 1: parity is forced based on EPS and PEN.	0x0	RW
4	EPS	Parity select. This bit only has meaning if parity is enabled (PEN set). 0: odd parity is transmitted and checked. 1: even parity is transmitted and checked.	0x0	RW
3	PEN	Parity enable. Used to control of the parity bit transmitted and checked. The value transmitted and the value checked is based on the settings of EPS and SP. 0: parity is not transmitted or checked. 1: parity is transmitted and checked.	0x0	RW
2	STOP	Stop bit. Used to control the number of stop bits transmitted. In all cases, only the first stop bit is evaluated on data received. 0: send one stop bit regardless of the word length (WLS). 1: send a number of stop bits based on the word length as follows: WLS = 00, 1.5 stop bits transmitted (5-bit word length); WLS = 01 or 10 or 11, 2 stop bits transmitted (6-, 7-, or 8-bit word length).	0x0	RW
[1:0]	WLS	Word length select. Selects the number of bits per transmission. 00: 5 bits. 01: 6 bits. 10: 7 bits. 11: 8 bits.	0x0	RW

Modem Control Register

Address: 0x40005010, Reset: 0x0000, Name: COMMCR

Table 310. Bit Descriptions for COMMCR

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	R
4	LOOPBACK	Loopback mode. In loopback mode, the SOUT is forced high. The modem signals are also directly connected to the status inputs (RTS to CTS, DTR to DSR, OUT1 to RI, and OUT2 to DCD). 0: normal operation, loopback disabled. 1: loopback enabled.	0x0	RW
3	OUT2	Output 2. 0: force OUT2 to a Logic 1. 1: force OUT2 to a Logic 0.	0x0	RW
2	OUT1	Output 1. 0: force OUT1 to a Logic 1. 1: force OUT1 to a Logic 0.	0x0	RW
1	RTS	Request to send. 0: force RTS to a Logic 1. 1: force RTS to a Logic 0.	0x0	RW
0	DTR	Data terminal ready. 0: force DTR to a Logic 1. 1: force DTR to a Logic 0.	0x0	RW

Line Status Register

Address: 0x40005014, Reset: 0x0060, Name: COMLSR

Table 311. Bit Descriptions for COMLSR

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	R
6	TEMT	COMTX and shift register empty status. 0: COMTX has been written to and contains data to be transmitted. Care must be taken not to overwrite its value. 1: COMTX and the transmit shift register are empty and it is safe to write new data to COMTX. Data has been transmitted.	0x1	R
5	THRE	COMTX empty. THRE is cleared when COMRX is read. 0: COMTX has been written to and contains data to be transmitted. Care must be taken not to overwrite its value. 1: COMTX is empty and it is safe to write new data to COMTX. The previous data may not have been transmitted yet and can still be present in the shift register.	0x1	R
4	BI	Break indicator. If set, this bit self clears after COMLSR is read. 0: SIN was not detected to be longer than the maximum word length. 1: SIN was held low for more than the maximum word length.	0x0	RC
3	FE	Framing error. If set, this bit self clears after COMLSR is read. 0: no invalid Stop bit was detected. 1: an invalid Stop bit was detected on a received word.	0x0	RC
2	PE	Parity error. If set, this bit self clears after COMLSR is read. 0: no parity error was detected. 1: a parity error occurred on a received word.	0x0	RC
1	OE	Overrun error. If set, this bit self clears after COMLSR is read. 0: receive data has not been overwritten. 1: receive data was overwritten by new data before COMRX was read.	0x0	RC
0	DR	Data ready. This bit is cleared only by reading COMRX. This bit does not self clear. 0: COMRX does not contain new receive data. 1: COMRX contains receive data that must be read.	0x0	RC

Modem Status Register

Address: 0x40005018, Reset: 0x0000, Name: COMMSR

Table 312. Bit Descriptions for COMMSR

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	DCD	Data carrier detect. This bit reflects the direct status complement of the DCD pin. 0: NDCD is currently logic high. 1: NDCD is currently logic low.	0x0	R
6	RI	Ring indicator. This bit reflects the direct status complement of the DCD pin. 0: NRI is currently logic high. 1: NRI is currently logic low.	0x0	R
5	DSR	Data set ready. This bit reflects the direct status complement of the DCD pin. 0: NDSR is currently logic high. 1: NDSR is currently logic low.	0x0	R
4	CTS	Clear to send. This bit reflects the direct status complement of the DCD pin. 0: NCTS is currently logic high. 1: NCTS is currently logic low.	0x0	R
3	DDCD	Delta DCD. If set, this bit self clears after COMMSR is read. 0: DCD has not changed state since COMMSR was last read. 1: DCD changed state since COMMSR last read.	0x0	R
2	TERI	Trailing edge RI. If set, this bit self clears after COMMSR is read. 0: RI has not changed from 0 to 1 since COMMSR was last read. 1: RI changed from 0 to 1 since COMMSR was last read.	0x0	R
1	DDSR	Delta DSR. If set, this bit self clears after COMMSR is read. 0: DSR has not changed state since COMMSR was last read. 1: DSR changed state since COMMSR was last read.	0x0	R
0	DCTS	Delta CTS. If set, this bit self clears after COMMSR is read. 0: CTS has not changed state since COMMSR was last read. 1: CTS changed state since COMMSR was last read.	0x0	R

Scratch Buffer Register

Address: 0x4000501C, Reset: 0x0000, Name: COMSCR

Table 313. Bit Descriptions for COMSCR

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	SCR	Scratch. The scratch register is an 8-bit register used to store intermediate results. The value contained in the scratch register does not affect UART functionality or performance. Only 8 bits of this register are implemented. Bits[15:8] are read only and always return 0x00 when read. Writable with any value from 0 to 255. A read returns the last value written.	0x0	RW

SOUT Modulation Configuration Register

Address: 0x40005020, Reset: 0x0000, Name: COMMCFG

Table 314. Bit Descriptions for COMMCFG

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
[9:0]	MODCLK_CFG	38 k modulation configuration.	0x0	RW

Fractional Baud Rate Register

Address: 0x40005024, Reset: 0x0000, Name: COMFBR

Table 315. Bit Descriptions for COMFBR

Bits	Bit Name	Description	Reset	Access
15	FBEN	Fractional baud rate generator enable. The generating of fractional baud rate can be described by the following formula and the final baud rate of UART operation is calculated as baud rate = (UCLK/(2 × (M + N/2048) × 16 × COMDIV)).	0x0	RW
[14:13]	RESERVED	Reserved.	0x0	R
[12:11]	DIVM	Fractional baud rate M divide, Bits[3:1]. This bit must not be 0.	0x0	RW
[10:0]	DIVN	Fractional baud rate N divide, Bits[2047:0].	0x0	RW

Baud Rate Divider Register

Address: 0x40005028, Reset: 0x0001, Name: COMDIV

Table 316. Bit Descriptions for COMDIV

Bits	Bit Name	Description	Reset	Access
[15:0]	DIV	Baud rate divider. The COMDIV register must not be 0, which is not specified. The range of allowed DIV values is from 1 to 65535.	0x1	RW

UART Control Register

Address: 0x40005030, Reset: 0x0000, Name: COMCTL

Table 317. Bit Descriptions for COMCTL

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0	R
0	DIS	UART disable control. 0: UART enable. 1: UART disable.	0x0	RW

GPIOs

FUNCTIONALITY

The ADuCM350 features up to 66 general-purpose bidirectional input/output (GPIO) pins. Most of the GPIO pins have multiple functions, configurable by user code, when multiplexed at the top level of an ADuCM350 product.

The GPIOs are grouped into five ports: P0, P1, P2, P3, and P4. P0, P1, P2, P3, and P4 contain 16, 16, 16, 15, and 3 GPIOs, respectively.

Each GPIO can be configured as an input or an output. The GPIOs also have internal pull-up or pull-down programmable resistors. All input/output pins are functional over the full supply range (VBAT = 1.8 V to 3.6 V), and the logic input voltages are specified as percentages of the supply.

$V_{INL} = 0.2 \times IOVDD_{MAX}$

$V_{INH} = 0.7 \times IOVDD_{MIN}$

Absolute maximum input voltage is IOVDD + 0.3 V.

When the ADuCM350 enters a power saving mode, the GPIO pins retain their state. Note that a driving peripheral cannot drive the pin. For example, if the UART is driving the pin on entry into deep sleep, it is isolated from the pin and power gated. Its state and control is then restored upon wake-up.

GPIO CONTROL

The ADuCM350 controls the GPIO pins through a set of MMR registers. These MMRs are implemented as part of an APB32 peripheral, starting at Base Address 0x40020000. Multiple functions are mapped on most of the GPIO pins. These functions can be selected by appropriately configuring the corresponding ports of the GPxCON control registers.

BLOCK DIAGRAM

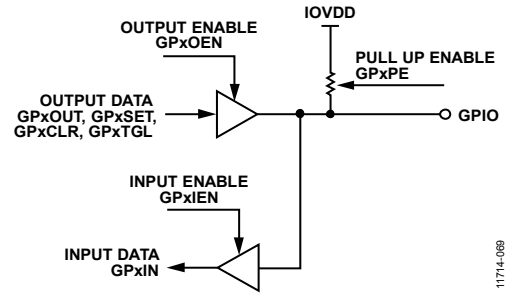


Figure 70. GPIO Pins with Pull-Up Resistor

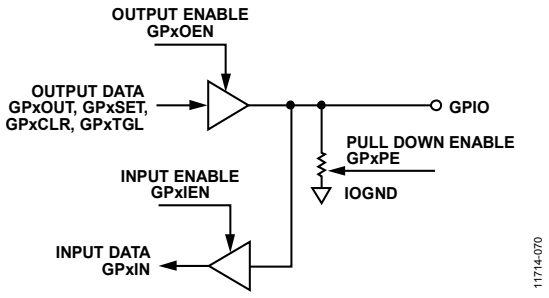


Figure 71. GPIO Pins with Pull-Down Resistor

OPERATION

IO Pull-Up or Pull-Down Enable

All GPIO pins can be grouped into two categories: GPIO pins with a pull-up resister and GPIO pins with a pull-down resistor. Each GPIO port has a corresponding GPxPE register. Using the GPxPE registers, it is possible to enable/disable pull-up/pull-down registers on the pins when they are configured as inputs.

IO Data In

When the GPIOs are configured as input using the GPxIEN register, the GPIO input levels are available in the GPxIN register.

IO Data Out

When the GPIOs are configured as outputs, the values in the GPxOUT register are reflected on the GPIOs.

Bit Set

Each GPIO port has a corresponding bit set register: GPxSET. Using the bit set register, it is possible to set one or more GPIO data outs without affecting others within the port. Only the GPIO corresponding with the write data bit equal to one is set, the remaining GPIOs are unaffected.

Bit Clear

Each GPIO port has a corresponding bit clear register: GPxCLR. Use the bit clear register to clear one or more GPIO data outs without affecting others within the port. Only the GPIO corresponding with the write data bit equal to one is cleared, the remaining GPIOs are unaffected.

Bit Toggle

Each GPIO port has a corresponding bit toggle register: GPxTGL. Using the bit toggle register, it is possible to invert one or more GPIO data outs without affecting others within the port. Only the GPIO corresponding with the write data bit equal to one are toggled, the remaining GPIOs are unaffected.

IO Data Output Enable

Each GPIO port has a data output enable (GPxOEN) register, by which the data output path is enabled. When the data output enable register bits are set, the values in GPxOUT are reflected on the corresponding GPIO pins.

Interrupts

Each GPIO pin can be associated with an interrupt. Interrupts can be independently enabled for each GPIO pin and are always edge detect; only one interrupt is generated with each GPIO pin transition. The polarity of the detected edge can be positive (low-to-high) or negative (high-to-low). Each GPIO interrupt event can be mapped to one of two interrupts (INTA or INTB). This allows the system more flexibility in terms of how GPIO interrupts are grouped for servicing and how interrupt priorities are set. The interrupt status of each GPIO pin can be determined and cleared by accessing the associated status registers.

Interrupt Polarity

The polarity of the interrupt determines whether the interrupt is accepted on the rising or falling edge. Each GPIO port has a corresponding interrupt register (GPxPOL) by which the interrupt polarity of each pin is configured. When GPxPOL is set to 0, an interrupt event is latched on a high to low transition on the corresponding pin. When GPxPOL is set to 1, an interrupt event is latched on a low-to-high transition on the corresponding pin.

Interrupt A Enable

Each GPIO port has a corresponding Interrupt A enable (GPxIENA) register that is enabled or masked for each pin in the port. These register bits determine whether a latched edge event is allowed to interrupt the core (Interrupt A) or is masked. In either case, the occurrence of the event is captured in the corresponding bit of the GPxINT status register. When GPxIENA is set to 0, Interrupt A is not enabled (masked), and no interrupts to the core are generated by this GPIO pin. When GPxIENA is set to 1, Interrupt A is enabled, and on a valid detected edge, an interrupt source to the core is generated.

Interrupt B Enable

Each GPIO port has a corresponding Interrupt B enable (GPxIENB) register that is enabled or masked for each pin in the port. These register bits determine whether a latched edge event is allowed to interrupt the core (Interrupt B) or is masked. In either case, the occurrence of the event is captured in the corresponding bit of the GPxINT status register. When GPxIENB is set to 0, Interrupt B is not enabled (masked), and no interrupts to the core are generated by this GPIO pin. When GPxIENB is set to 1, Interrupt B is enabled, and on a valid detected edge, an interrupt source to the core is generated.

Interrupt Status

Each GPIO port has an interrupt status register (GPxINT) that captures the interrupts occurring on its pins. These register bits indicate that the appropriately configured rising or falling edge has been detected on the corresponding GPIO pin. After an event is detected, GPxINT remains set until cleared; this is true even if the GPIO pin transitions back to a nonactive state. Out of reset pull-ups combined with falling edge detect can result in the GPxINT status being cleared; however, this may not be the case based on pin activity. It is, therefore, recommended that the status of GPxINT be checked before enabling interrupts (GPXIENA and GPXIENB) initially as well as any time the GPIO pins are configured. Interrupt bits are cleared by writing a 1 to the appropriate bit location. Writing a 0 has no effect. If interrupts are enabled to the core (GPxIENA, GPxIENB), an INT (GPxINTA or GPxINTB) value of 1 results in an interrupt to the core. This bit must be cleared during servicing of the interrupt.

When read as 0, a rising or falling edge was not detected on the corresponding GPIO pin because this bit was last cleared. When read as 1, a rising or falling edge (POL selectable) was detected on the corresponding GPIO pin. This bit can be software cleared by writing a 1 to this bit location. This bit can only be subsequently set again after the pin returns to its nonasserted state and then returns to an asserted state.

Drive Strength

The drive strength of the P1.X pins can be increased by enabling the respective bit in the GP1DS register. To increase the strength on P1.7, set Bit 7 to 1. Drive strength performance is highlighted in the ADuCM350 data sheet.

SYSTEM CLOCKS

The ADuCM350 provides the ability to drive the system clock externally. The GPIO pin P0.11 is used for this purpose. The system can be clocked using P0.11 after the input path for this pin is enabled by setting Bit 11 of the GP0IEN register.

The ADuCM350 provides the facility to observe the various clocks used in the system on an external pin. The GPIO pin P1.7 is used for this purpose. This feature is available as Function 4 on the P1.7 pin. To observe this clock on P1.7, select the function using the GP1CON registers (GP1CON[15:14] = 0x3). The clock source that appears on P1.7 is selected in the miscellaneous clock settings register (Bits[7:4] of the CLKCON0 register). See the System Clocks section for more details.

GPIO MEMORY MAPPED REGISTERS

GPIO Register Map

Table 318. GPIO Register Summary

Address	Name	Description	Reset	RW
0x40020000	GP0CON	GPIO Port 0 configuration	0x00000000	RW
0x40020004	GP0OEN	GPIO Port 0 output enable	0x0000	RW
0x40020008	GP0PE	GPIO Port 0 output pull-up/pull-down enable	0x03C0	RW
0x4002000C	GP0IEN	GPIO Port 0 input path enable	0x0000	RW
0x40020010	GP0IN	GPIO Port 0 registered data input	0xXXXX	R
0x40020014	GP0OUT	GPIO Port 0 data output	0x0000	RW
0x40020018	GP0SET	GPIO Port 0 data out set	0x0000	W
0x4002001C	GP0CLR	GPIO Port 0 data out clear	0x0000	W
0x40020020	GP0TGL	GPIO Port 0 pin toggle	0x0000	W
0x40020024	GP0POL	GPIO Port 0 interrupt polarity	0x0000	RW
0x40020028	GP0IENA	GPIO Port 0 Interrupt A enable	0x0000	RW
0x4002002C	GP0IENB	GPIO Port 0 Interrupt B enable	0x0000	RW
0x40020030	GP0INT	GPIO Port 0 interrupt status	0x0000	RW1C
0x40020040	GP1CON	GPIO Port 1 configuration	0x00000000	RW
0x40020044	GP1OEN	GPIO Port 1 output enable	0x0000	RW
0x40020048	GP1PE	GPIO Port 1 output pull-up/pull-down enable	0x0000	RW
0x4002004C	GP1IEN	GPIO Port 1 input path enable	0x0000	RW
0x40020050	GP1IN	GPIO Port 1 registered data input	0x0000	R
0x40020054	GP1OUT	GPIO Port 1 data output	0x0000	RW
0x40020058	GP1SET	GPIO Port 1 data out set	0x0000	W
0x4002005C	GP1CLR	GPIO Port 1 data out clear	0x0000	W
0x40020060	GP1TGL	GPIO Port 1 pin toggle	0x0000	W
0x40020064	GP1POL	GPIO Port 1 interrupt polarity	0x0000	RW
0x40020068	GP1IENA	GPIO Port 1 Interrupt A enable	0x0000	RW
0x4002006C	GP1IENB	GPIO Port 1interrupt B enable	0x0000	RW
0x40020070	GP1INT	GPIO Port 1 interrupt status	0x0000	RW1C
0x40020074	GP1DS	GPIO Port 1 drive strength select	0x0000	RW
0x40020080	GP2CON	GPIO Port 2 configuration	0x00000000	RW
0x40020084	GP2OEN	GPIO Port 2 output enable	0x0000	RW
0x40020088	GP2PE	GPIO Port 2 output pull-up/pull-down enable	0x0000	RW
0x4002008C	GP2IEN	GPIO Port 2 input path enable	0x0000	RW
0x40020090	GP2IN	GPIO Port 2 registered data input	0x0000	R
0x40020094	GP2OUT	GPIO Port 2 data output	0x0000	RW

Address	Name	Description	Reset	RW
0x40020098	GP2SET	GPIO Port 2 data out set	0x0000	W
0x4002009C	GP2CLR	GPIO Port 2 data out clear	0x0000	W
0x400200A0	GP2TGL	GPIO Port 2 pin toggle	0x0000	W
0x400200A4	GP2POL	GPIO Port 2 interrupt polarity	0x0000	RW
0x400200A8	GP2IENA	GPIO Port 2 Interrupt A enable	0x0000	RW
0x400200AC	GP2IENB	GPIO Port 2 Interrupt B enable	0x0000	RW
0x400200B0	GP2INT	GPIO Port 2 interrupt status	0x0000	RW1C
0x400200C0	GP3CON	GPIO Port 3 configuration	0x00000000	RW
0x400200C4	GP3OEN	GPIO Port 3 output enable	0x0000	RW
0x400200C8	GP3PE	GPIO Port 3 output pull-up/pull-down enable	0x0000	RW
0x400200CC	GP3IEN	GPIO Port 3 input enable	0x0000	RW
0x400200D0	GP3IN	GPIO Port 3 registered data input	0x0000	R
0x400200D4	GP3OUT	GPIO Port 3 data output	0x0000	RW
0x400200D8	GP3SET	GPIO Port 3 data out set	0x0000	W
0x400200DC	GP3CLR	GPIO Port 3 data out clear	0x0000	W
0x400200E0	GP3TGL	GPIO Port 3 pin toggle	0x0000	W
0x400200E4	GP3POL	GPIO Port 3 interrupt polarity	0x0000	RW
0x400200E8	GP3IENA	GPIO Port 3 Interrupt A enable	0x0000	RW
0x400200EC	GP3IENB	GPIO Port 3 Interrupt B enable	0x0000	RW
0x400200F0	GP3INT	GPIO Port 3 interrupt status	0x0000	RW1C
0x40020100	GP4CON	GPIO Port 4 configuration	0x00000000	RW
0x40020104	GP4OEN	GPIO Port 4 output enable	0x0000	RW
0x40020108	GP4PE	GPIO Port 4 output pull-up/pull-down enable	0x0000	RW
0x4002010C	GP4IEN	GPIO Port 4 input path enable	0x0000	RW
0x40020110	GP4IN	GPIO Port 4 registered data input	0x0000	R
0x40020114	GP4OUT	GPIO Port 4 data output	0x0000	RW
0x40020118	GP4SET	GPIO Port 4 data out set	0x0000	W
0x4002011C	GP4CLR	GPIO Port 4 data out clear	0x0000	W
0x40020120	GP4TGL	GPIO Port 4 pin toggle	0x0000	W
0x40020124	GP4POL	GPIO Port 4 interrupt polarity	0x0000	RW
0x40020128	GP4IENA	GPIO Port 4 Interrupt A enable	0x0000	RW
0x4002012C	GP4IENB	GPIO Port 4 Interrupt B enable	0x0000	RW
0x40020130	GP4INT	GPIO Port 4 interrupt status	0x0000	RW1C

GPIO Port 0 Configuration Register

Address: 0x40020000, Reset: 0x00000000, Name: GP0CON

The GP0CON register is reserved for top level pin muxing for the GPIO block.

Note that any combination not documented are reserved and must not be used.

Table 319. Bit Descriptions for GP0CON

Bits	Bit Name	Description	Reset	Access
[31:30]	PIN15_CFG	p0.15 configuration bits. 00: GPIO. 01: SPIH chip select.	0x0	RW
[29:28]	PIN14_CFG	p0.14 configuration bits. 00: GPIO. 01: SPIH MOSI.	0x0	RW
[27:26]	PIN13_CFG	p0.13 configuration bits. 00: GPIO. 01: SPIH MISO.	0x0	RW
[25:24]	PIN12_CFG	p0.12 configuration bits. 00: GPIO. 01: SPIH SCLK.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[23:22]	PIN11_CFG	p0.11 configuration bits. 00: GPIO.	0x0	RW
[21:20]	PIN10_CFG	p0.10 configuration bits. 00: GPIO. 01: Timer C PWM output.	0x0	RW
[19:18]	PIN9_CFG	p0.9 configuration bits. 00: JTAG clock/serial wire clock. 01: GPIO.	0x0	RW
[17:16]	PIN8_CFG	p0.8 configuration bits. 00: JTAG test mode select/serial wire data. 01: GPIO.	0x0	RW
[15:14]	PIN7_CFG	p0.7 configuration bits. 00: JTAG serial data input. 01: GPIO. 10: UART Rx.	0x0	RW
[13:12]	PIN6_CFG	p0.6 configuration bits. 00: JTAG serial data out/serial wire trace out. 01: GPIO. 10: UART Tx.	0x0	RW
[11:10]	PIN5_CFG	p0.5 configuration bits. 00: GPIO. 01: CapTouch F down button.	0x0	RW
[9:8]	PIN4_CFG	p0.4 configuration bits. 00: GPIO. 01: CapTouch E up button.	0x0	RW
[7:6]	PIN3_CFG	p0.3 configuration bits. 00: GPIO. 01: CapTouch D enter button.	0x0	RW
[5:4]	PIN2_CFG	p0.2 configuration bits. 00: GPIO. 01: CapTouch C right button.	0x0	RW
[3:2]	PIN1_CFG	p0.1 configuration bits. 00: GPIO. 01: CapTouch B left button.	0x0	RW
[1:0]	PIN0_CFG	p0.0 configuration bits. 00: GPIO. 01: CapTouch A on/off button.	0x0	RW

GPIO Port 0 Output Enable Register

Address: 0x40020004, Reset: 0x0000, Name: GP0OEN

Table 320. Bit Descriptions for GP0OEN

Bits	Bit Name	Description	Reset	Access
[15:0]	OEN	Pin output drive enable. Each bit is set to enable the output for that particular pin. It is cleared to disable the output for each pin.	0x0000	RW

GPIO Port 0 Output Pull-Up/Pull-Down Enable Register

Address: 0x40020008, Reset: 0x03C0, Name: GP0PE

Table 321. Bit Descriptions for GP0PE

Bits	Bit Name	Description	Reset	Access
[15:0]	PE	Pin pull enable. Each bit is set to enable the pull-up/pull-down for that particular pin. It is cleared to disable the pull-up/pull-down for each pin.	0x03c0	RW

GPIO Port 0 Input Path Enable Register

Address: 0x4002000C, Reset: 0x0000, Name: GP0IEN

Table 322. Bit Descriptions for GP0IEN

Bits	Bit Name	Description	Reset	Access
[15:0]	IEN	Input path enable. Each bit is set to enable the input path and cleared to disable the input path for the GPIO pin.	0x0000	RW

GPIO Port 0 Registered Data Input

Address: 0x40020010, Reset: 0x000X, Name: GP0IN

Table 323. Bit Descriptions for GP0IN

Bits	Bit Name	Description	Reset	Access
[15:0]	IN	Registered data input. Each bit reflects the state of the GPIO pin if the corresponding input buffer is enabled. If the pin input buffer is disabled the value seen is zero.	0xx	R

GPIO Port 0 Data Output Register

Address: 0x40020014, Reset: 0x0000, Name: GP0OUT

Table 324. Bit Descriptions for GP0OUT

Bits	Bit Name	Description	Reset	Access
[15:0]	OUT	Data out. Set by user code to drive the corresponding GPIO high. Cleared by user to drive the corresponding GPIO low.	0x0000	RW

GPIO Port 0 Data Out Set Register

Address: 0x40020018, Reset: 0x0000, Name: GP0SET

Table 325. Bit Descriptions for GP0SET

Bits	Bit Name	Description	Reset	Access
[15:0]	SET	Set the output high for the pin. Set by user code to drive the corresponding GPIO high. Clearing this bit has no effect.	0x0000	W

GPIO Port 0 Data Out Clear Register

Address: 0x4002001C, Reset: 0x0000, Name: GP0CLR

Table 326. Bit Descriptions for GP0CLR

Bits	Bit Name	Description	Reset	Access
[15:0]	CLR	Set the output low for the port pin. Each bit is set to drive the corresponding GPIO pin low. Clearing this bit has no effect.	0x0000	W

GPIO Port 0 Pin Toggle Register

Address: 0x40020020, Reset: 0x0000, Name: GP0TGL

Table 327. Bit Descriptions for GP0TGL

Bits	Bit Name	Description	Reset	Access
[15:0]	TGL	Toggle the output of the port pin. Each bit is set to invert the corresponding GPIO pin. Clearing this bit has not effect.	0x0000	W

GPIO Port 0 Interrupt Polarity Register

Address: 0x40020024, Reset: 0x0000, Name: GP0POL

Table 328. Bit Descriptions for GP0POL

Bits	Bit Name	Description	Reset	Access
[15:0]	INTPOL	Interrupt polarity. Determines whether interrupts are generated on the rising or falling edge of the corresponding GPIO pin. When cleared, an interrupt event is latched on a high to low transition. When set, an interrupt event is latched on a low to high transition.	0x0000	RW

GPIO Port 0 Interrupt A Enable Register

Address: 0x40020028, Reset: 0x0000, Name: GP0IENA

Table 329. Bit Descriptions for GP0IENA

Bits	Bit Name	Description	Reset	Access
[15:0]	INTAEN	Interrupt A enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt A) or be masked. In either case, the occurrence of the event is captured in the GP0INT status register. when cleared Interrupt A is not enabled (masked). When set Interrupt A is enabled.	0x0000	RW

GPIO Port 0 Interrupt B Enable Register

Address: 0x4002002C, Reset: 0x0000, Name: GP0IENB

Table 330. Bit Descriptions for GP0IENB

Bits	Bit Name	Description	Reset	Access
[15:0]	INTBEN	Interrupt B enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt B) or be masked. In either case, the occurrence of the event is captured in the GP0INT status register. When set to 0 Interrupt B is not enabled (masked). When set to 1 Interrupt A is enabled.	0x0000	RW

GPIO Port 0 Interrupt Status Register

Address: 0x40020030, Reset: 0x0000, Name: GP0INT

Table 331. Bit Descriptions for GP0INT

Bits	Bit Name	Description	Reset	Access
[15:0]	INTSTATUS	Interrupt status. Indicates that the appropriately configured rising or falling edge has been detected on the corresponding GPIO pin. After an event is detected, the INTSTATUS bit remain set until cleared, this is true even if the GPIO pin transitions back to a nonactive state. INTSTATUS bits are cleared by writing 1 to the appropriate bit location. Writing 0 has no effect.	0x0000	RW1C

GPIO Port 1 Configuration Register

Address: 0x40020040, Reset: 0x00000000, Name: GP1CON

The GP1CON register is reserved for top-level pin muxing for the GPIO block.

Note that any combination not documented is reserved and must not be used.

Table 332. Bit Descriptions for GP1CON

Bits	Bit Name	Description	Reset	Access
[31:30]	PIN_15_CFG	p1.15 configuration bits. 00: GPIO. 01: LCD Driver Segment 18. 10: PDI Data 15.	0x0	RW
[29:28]	PIN_14_CFG	p1.14 configuration bits. 00: GPIO. 01: LCD Driver Segment 17. 10: PDI Data 14.	0x0	RW
[27:26]	PIN13_CFG	p1.13 configuration bits. 00: GPIO. 01: LCD Driver Segment 16. 10: PDI Data 13.	0x0	RW
[25:24]	PIN12_CFG	p1.12 configuration bits. 00: GPIO. 01: LCD Driver Segment 15. 10: PDI Data 12.	0x0	RW
[23:22]	PIN11_CFG	p1.11 configuration bits. 00: GPIO. 01: LCD Driver Segment 14. 10: PDI Data 11.	0x0	RW
[21:20]	PIN10_CFG	p1.10 configuration bits. 00: GPIO. 01: LCD Driver Segment 13. 10: PDI Data 10.	0x0	RW
[19:18]	PIN9_CFG	p1.9 configuration bits. 00: GPIO. 01: LCD Driver Segment 12. 10: PDI Data 9.	0x0	RW
[17:16]	PIN8_CFG	p1.8 configuration bits. 00: GPIO. 01: LCD Driver Segment 11. 10: PDI Data 8.	0x0	RW
[15:14]	PIN7_CFG	p1.7 configuration bits. 00: GPIO. 01: LCD Driver Segment 10. 10: PDI Data 7. 11: System clock out.	0x0	RW
[13:12]	PIN6_CFG	p1.6 configuration bits. 00: GPIO. 01: LCD Driver Segment 9. 10: PDI Data 6.	0x0	RW
[11:10]	PIN5_CFG	p1.5 configuration bits. 00: GPIO. 01: LCD Driver Segment 8. 10: PDI Data 5.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[9:8]	PIN4_CFG	p1.4 configuration bits. 00: GPIO. 01: LCD Driver Segment 7. 10: PDI Data 4.	0x0	RW
[7:6]	PIN3_CFG	p1.3 configuration bits. 00: GPIO. 01: LCD Driver Segment 6 10: PDI Data 3.	0x0	RW
[5:4]	PIN2_CFG	p1.2 configuration bits. 00: GPIO. 01: LCD Driver Segment 5. 10: PDI Data 2. 11: PDI serial data in.	0x0	RW
[3:2]	PIN1_CFG	p1.1 configuration bits. 00: GPIO. 01: LCD Driver Segment 4. 10: PDI Data 1. 11: PDI serial data out.	0x0	RW
[1:0]	PIN0_CFG	p1.0 configuration bits. 00: GPIO. 01: LCD Driver Segment 3. 10: PDI Data 0. 11: PDI serial clock.	0x0	RW

GPIO Port 1 Output Enable Register

Address: 0x40020044, Reset: 0x0000, Name: GP1OEN

Table 333. Bit Descriptions for GP1OEN

Bits	Bit Name	Description	Reset	Access
[15:0]	OEN	Pin output drive enable. Each bit is set to enable the output for that particular pin. It is cleared to disable the output for each pin.	0x0000	RW

GPIO Port 1 Output Pull-Up/Pull-Down Enable Register

Address: 0x40020048, Reset: 0x0000, Name: GP1PE

Table 334. Bit Descriptions for GP1PE

Bits	Bit Name	Description	Reset	Access
[15:0]	PE	Pin pull enable. Each bit is set to enable the pull-up/pull-down for that particular pin. It is cleared to disable the pull-up/pull-down for each pin.	0x0000	RW

GPIO Port 1 Input Path Enable Register

Address: 0x4002004C, Reset: 0x0000, Name: GP1IEN

Table 335. Bit Descriptions for GP1IEN

Bits	Bit Name	Description	Reset	Access
[15:0]	IEN	Input path enable. Each bit is set to enable the input path and cleared to disable the input path for the GPIO pin.	0x0000	RW

GPIO Port 1 Registered Data Input

Address: 0x40020050, Reset: 0x0000, Name: GP1IN

Table 336. Bit Descriptions for GP1IN

Bits	Bit Name	Description	Reset	Access
[15:0]	IN	Registered data input. Each bit reflects the state of the GPIO pin if the corresponding input buffer is enabled. If the pin input buffer is disabled the value seen is zero.	0x0000	R

GPIO Port 1 Data Output Register

Address: 0x40020054, Reset: 0x0000, Name: GP1OUT

Table 337. Bit Descriptions for GP1OUT

Bits	Bit Name	Description	Reset	Access
[15:0]	OUT	Data out. Set by user code to drive the corresponding GPIO high. Cleared by user to drive the corresponding GPIO low.	0x0000	RW

GPIO Port 1 Data Out Set Register

Address: 0x40020058, Reset: 0x0000, Name: GP1SET

Table 338. Bit Descriptions for GP1SET

Bits	Bit Name	Description	Reset	Access
[15:0]	SET	Set the output high for the pin. Set by user code to drive the corresponding GPIO high. Clearing this bit has no effect.	0x0000	W

GPIO Port 1 Data Out Clear Register

Address: 0x4002005C, Reset: 0x0000, Name: GP1CLR

Table 339. Bit Descriptions for GP1CLR

Bits	Bit Name	Description	Reset	Access
[15:0]	CLR	Set the output low for the port pin. Each bit is set to drive the corresponding GPIO pin low. Clearing this bit has no effect.	0x0000	W

GPIO Port 1 Pin Toggle Register

Address: 0x40020060, Reset: 0x0000, Name: GP1TGL

Table 340. Bit Descriptions for GP1TGL

Bits	Bit Name	Description	Reset	Access
[15:0]	TGL	Toggle the output of the port pin. Each bit is set to invert the corresponding GPIO pin. Clearing this bit has no effect.	0x0000	W

GPIO Port 1 Interrupt Polarity Register

Address: 0x40020064, Reset: 0x0000, Name: GP1POL

Table 341. Bit Descriptions for GP1POL

Bits	Bit Name	Description	Reset	Access
[15:0]	INTPOL	Interrupt polarity. Determines whether interrupts are generated on the rising or falling edge of the corresponding GPIO pin. When cleared, an interrupt event is latched on a high to low transition. When set, an interrupt event is latched on a low to high transition.	0x0000	RW

GPIO Port 1 Interrupt A Enable Register

Address: 0x40020068, Reset: 0x0000, Name: GP1IENA

Table 342. Bit Descriptions for GP1IENA

Bits	Bit Name	Description	Reset	Access
[15:0]	INTAEN	Interrupt A enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt A) or be masked. In either case, the occurrence of the event is captured in the GP1INT status register. When cleared, Interrupt A is not enabled (masked). When set Interrupt A is enabled.	0x0000	RW

GPIO Port 1 Interrupt B Enable Register

Address: 0x4002006C, Reset: 0x0000, Name: GP1IENB

Table 343. Bit Descriptions for GP1IENB

Bits	Bit Name	Description	Reset	Access
[15:0]	INTBEN	Interrupt B enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt B) or be masked. In either case, the occurrence of the event is captured in the GP1INT status register. When cleared, Interrupt B is not enabled (masked). When set, Interrupt B is enabled.	0x0000	RW

GPIO Port 1 Interrupt Status Register

Address: 0x40020070, Reset: 0x0000, Name: GP1INT

Table 344. Bit Descriptions for GP1INT

Bits	Bit Name	Description	Reset	Access
[15:0]	INTSTATUS	Interrupt status. Indicates that the appropriately configured rising or falling edge has been detected on the corresponding GPIO pin. After an event is detected, the INTSTATUS bit remains set until cleared. This is true even if the GPIO pin transitions back to a nonactive state. INTSTATUS bits are cleared by writing 1 to the appropriate bit location. Writing 0 has no effect.	0x0000	RW1C

GPIO Port 1 Drive Strength Select Register

Address: 0x40020074, Reset: 0x0000, Name: GP1DS

The GPI1DS register is used to set the drive strength of the GPIO1 pins.

Table 345. Bit Descriptions for GP1DS

Bits	Bit Name	Description	Reset	Access
[15:0]	DS	Drive strength select. Each bit is configured to set the drive strength of the corresponding GPIO pin.	0x0	RW

GPIO Port 2 Configuration Register

Address: 0x40020080, Reset: 0x00000000, Name: GP2CON

The GP2CON register is reserved for top level pin muxing for the GPIO block.

Note that any combination not documented is reserved and must not be used.

Table 346. Bit Descriptions for GP2CON

Bits	Bit Name	Description	Reset	Access
[31:30]	PIN_15_CFG	P2.15 configuration bits. 00: GPIO. 01: LCD Driver Segment 28.	0x0	RW
[29:28]	PIN_14_CFG	P2.14 configuration bits. 00: GPIO. 01: LCD Driver Segment 27.	0x0	RW
[27:26]	PIN13_CFG	P2.13 configuration bits. 00: GPIO. 01: LCD Driver Segment 26.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[25:24]	PIN12_CFG	P2.12 configuration bits. 00: GPIO. 01: LCD Driver Segment 25.	0x0	RW
[23:22]	PIN11_CFG	P2.11 configuration bits. 00: GPIO. 01: LCD Driver Segment 24.	0x0	RW
[21:20]	PIN10_CFG	P2.10 configuration bits. 00: GPIO. 01: LCD Driver Segment 23.	0x0	RW
[19:18]	PIN9_CFG	P2.9 configuration bits. 00: GPIO. 01: LCD Driver Segment 22.	0x0	RW
[17:16]	PIN8_CFG	P2.8 configuration bits. 00: GPIO. 01: LCD Driver Segment 11.	0x0	RW
[15:14]	PIN7_CFG	P2.7 configuration bits. 00: GPIO. 01: LCD Driver Segment 20. 10: Timer A PWM output.	0x0	RW
[13:12]	PIN6_CFG	P2.6 configuration bits. 00: GPIO. 01: LCD Driver Segment 19. 10: PDI tearing effect.	0x0	RW
[11:10]	PIN5_CFG	P2.5 configuration bits. 00: GPIO. 01: LCD Driver Segment 2. 10: PDI E clock or WRX.	0x0	RW
[9:8]	PIN4_CFG	P2.4 configuration bits. 00: GPIO. 01: LCD Driver Segment 1. 10: PDI R/WX or RDX.	0x0	RW
[7:6]	PIN3_CFG	P2.3 configuration bits. 00: GPIO. 01: LCD Driver Back Plane 3. 10: PDI data/command.	0x0	RW
[5:4]	PIN2_CFG	P2.2 configuration bits. 00: GPIO. 01: LCD Driver Back Plane 2. 10: PDI chip select.	0x0	RW
[3:2]	PIN1_CFG	P2.1 configuration bits. 00: GPIO. 01: LCD Driver Back Plane 1. 10: PDI reset.	0x0	RW
[1:0]	PIN0_CFG	P2.0 configuration bits. 00: GPIO. 01: LCD Driver Back Plane 0.	0x0	RW

GPIO Port 2 Output Enable Register

Address: 0x40020084, Reset: 0x0000, Name: GP2OEN

Table 347. Bit Descriptions for GP2OEN

Bits	Bit Name	Description	Reset	Access
[15:0]	OEN	Pin output drive enable. Each bit is set to enable the output for that particular pin. It is cleared to disable the output for each pin.	0x0000	RW

GPIO Port 2 Output Pull-Up/Pull-Down Enable Register

Address: 0x40020088, Reset: 0x0000, Name: GP2PE

Table 348. Bit Descriptions for GP2PE

Bits	Bit Name	Description	Reset	Access
[15:0]	PE	Pin pull enable. Each bit is set to enable the pull-up/pull-down for that particular pin. It is cleared to disable the pull-up/pull-down for each pin.	0x0000	RW

GPIO Port 2 Input Path Enable Register

Address: 0x4002008C, Reset: 0x0000, Name: GP2IEN

Table 349. Bit Descriptions for GP2IEN

Bits	Bit Name	Description	Reset	Access
[15:0]	IEN	Input path enable. Each bit is set to enable the input path and cleared to disable the input path for the GPIO pin.	0x0000	RW

GPIO Port 2 Registered Data Input

Address: 0x40020090, Reset: 0x0000, Name: GP2IN

Table 350. Bit Descriptions for GP2IN

Bits	Bit Name	Description	Reset	Access
[15:0]	IN	Registered data input. Each bit reflects the state of the GPIO pin if the corresponding input buffer is enabled. If the pin input buffer is disabled the value seen is zero.	0x0000	R

GPIO Port 2 Data Output Register

Address: 0x40020094, Reset: 0x0000, Name: GP2OUT

Table 351. Bit Descriptions for GP2OUT

Bits	Bit Name	Description	Reset	Access
[15:0]	OUT	Data out. Set by user code to drive the corresponding GPIO high. Cleared by user to drive the corresponding GPIO low.	0x0000	RW

GPIO Port 2 Data Out Set Register

Address: 0x40020098, Reset: 0x0000, Name: GP2SET

Table 352. Bit Descriptions for GP2SET

Bits	Bit Name	Description	Reset	Access
[15:0]	SET	Set the output high for the pin. Set by user code to drive the corresponding GPIO high. Clearing this bit has no effect.	0x0000	W

GPIO Port 2 Data Out Clear Register

Address: 0x4002009C, Reset: 0x0000, Name: GP2CLR

Table 353. Bit Descriptions for GP2CLR

Bits	Bit Name	Description	Reset	Access
[15:0]	CLR	Set the output low for the port pin. Each bit is set to drive the corresponding GPIO pin low. Clearing this bit has no effect.	0x0000	W

GPIO Port 2 Pin Toggle Register

Address: 0x400200A0, Reset: 0x0000, Name: GP2TGL

Table 354. Bit Descriptions for GP2TGL

Bits	Bit Name	Description	Reset	Access
[15:0]	TGL	Toggle the output of the port pin. Each bit is set to invert the corresponding GPIO pin. Clearing this bit has not effect.	0x0000	W

GPIO Port 2 Interrupt Polarity Register

Address: 0x400200A4, Reset: 0x0000, Name: GP2POL

Table 355. Bit Descriptions for GP2POL

Bits	Bit Name	Description	Reset	Access
[15:0]	INTPOL	Interrupt polarity. Determines whether interrupts are generated on the rising or falling edge of the corresponding GPIO pin. When cleared, an interrupt event is latched on a high to low transition. When set, an interrupt event is latched on a low to high transition.	0x0000	RW

GPIO Port 2 Interrupt A Enable Register

Address: 0x400200A8, Reset: 0x0000, Name: GP2IENA

Table 356. Bit Descriptions for GP2IENA

Bits	Bit Name	Description	Reset	Access
[15:0]	INTAEN	Interrupt A enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt A) or be masked. In either case, the occurrence of the event is captured in the GP2INT status register. When cleared Interrupt A is not enabled (masked). When set, Interrupt A is enabled.	0x0000	RW

GPIO Port 2 Interrupt B Enable Register

Address: 0x400200AC, Reset: 0x0000, Name: GP2IENB

Table 357. Bit Descriptions for GP2IENB

Bits	Bit Name	Description	Reset	Access
[15:0]	INTBEN	Interrupt B enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt B) or be masked. In either case, the occurrence of the event is captured in the GP2INT status register. When cleared, Interrupt B is not enabled (masked). When set, Interrupt B is enabled.	0x0000	RW

GPIO Port 2 Interrupt Status Register

Address: 0x400200B0, Reset: 0x0000, Name: GP2INT

Table 358. Bit Descriptions for GP2INT

Bits	Bit Name	Description	Reset	Access
[15:0]	INTSTATUS	Interrupt status. Indicates that the appropriately configured rising or falling edge has been detected on the corresponding GPIO pin. After an event is detected, the INTSTATUS bit remains set until cleared. This is true even if the GPIO pin transitions back to a nonactive state. INTSTATUS bits are cleared by writing 1 to the appropriate bit location. Writing 0 has no effect.	0x0000	RW1C

GPIO Port 3 Configuration Register

Address: 0x400200C0, Reset: 0x00000000, Name: GP3CON

The GP3CON register is reserved for top level pin muxing for the GPIO block.

Note that any combination not documented is reserved and must not be used.

Table 359. Bit Descriptions for GP3CON

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	GPIO pin mux configuration.	0x0	R
[29:28]	PIN_14_CFG	P3.14 configuration bits. 00: GPIO. 10: I^2S LR clock.	0x0	RW
[27:26]	PIN13_CFG	P3.13 configuration bits. 00: GPIO. 01: Beeper Tone N. 10: I^2S data line.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[25:24]	PIN12_CFG	P3.12 configuration bits. 00: GPIO. 01: Beeper Tone P. 10: I²S bit clock.	0x0	RW
[23:22]	PIN11_CFG	P3.11 configuration bits. 00: GPIO. 01: LCD Driver Segment 32.	0x0	RW
[21:20]	PIN10_CFG	P3.10 configuration bits. 00: GPIO. 01: LCD Driver Segment 31.	0x0	RW
[19:18]	PIN9_CFG	P3.9 configuration bits. 00: GPIO. 01: LCD Driver Segment 30.	0x0	RW
[17:16]	PIN8_CFG	P3.8 configuration bits. 00: GPIO. 01: LCD Driver Segment 29.	0x0	RW
[15:14]	PIN7_CFG	P3.7 configuration bits. 00: GPIO. 01: UART Rx. 10: Timer C PWM output. 11: SPI1 chip select.	0x0	RW
[13:12]	PIN6_CFG	P3.6 configuration bits. 00: GPIO. 01: UART Tx. 10: Timer B PWM output. 11: SPI1 MOSI.	0x0	RW
[11:10]	PIN5_CFG	P3.5 configuration bits. 00: GPIO. 01: I²C SDA line. 11: SPI1MISO.	0x0	RW
[9:8]	PIN4_CFG	P3.4 configuration bits. 00: GPIO. 01: I²C SCL line. 11: SPI1 SCLK.	0x0	RW
[7:6]	PIN3_CFG	P3.3 configuration bits. 00: GPIO. 01: SPI0 chip select.	0x0	RW
[5:4]	PIN2_CFG	P3.2 configuration bits. 00: GPIO. 01: SPI0 MOSI.	0x0	RW
[3:2]	PIN1_CFG	P3.1 configuration bits. 00: GPIO. 01: SPI0 MISO.	0x0	RW
[1:0]	PIN0_CFG	P3.0 configuration bits. 00: GPIO. 01: SPI0 SCLK.	0x0	RW

GPIO Port 3 Output Enable Register

Address: 0x400200C4, Reset: 0x0000, Name: GP3OEN

Table 360. Bit Descriptions for GP3OEN

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	OEN	Pin output drive enable. Each bit is set to enable the output for that particular pin. It is cleared to disable the output for each pin.	0x000	RW

GPIO Port 3 Output Pull-Up/Pull-Down Enable Register

Address: 0x400200C8, Reset: 0x0000, Name: GP3PE

Table 361. Bit Descriptions for GP3PE

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	PE	Pin pull enable. Each bit is set to enable the pull-up/pull-down for that particular pin. It is cleared to disable the pull-up/pull-down for each pin.	0x000	RW

GPIO Port 3 Input Enable Register

Address: 0x400200CC, Reset: 0x0000, Name: GP3IEN

Table 362. Bit Descriptions for GP3IEN

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	IEN	Input path enable. Each bit is set to enable the input path and cleared to disable the input path for the GPIO pin.	0x000	RW

GPIO Port 3 Registered Data Input

Address: 0x400200D0, Reset: 0x0000, Name: GP3IN

Table 363. Bit Descriptions for GP3IN

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	IN	Registered data input. Each bit reflects the state of the GPIO pin if the corresponding input buffer is enabled. If the pin input buffer is disabled the value seen is zero.	0x000	R

GPIO Port 3 Data Output Register

Address: 0x400200D4, Reset: 0x0000, Name: GP3OUT

Table 364. Bit Descriptions for GP3OUT

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	OUT	Data out. Set by user code to drive the corresponding GPIO high. Cleared by user to drive the corresponding GPIO low.	0x000	RW

GPIO Port 3 Data Out Set Register

Address: 0x400200D8, Reset: 0x0000, Name: GP3SET

Table 365. Bit Descriptions for GP3SET

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	SET	Set the output high for the pin. Set by user code to drive the corresponding GPIO high. Clearing this bit has no effect.	0x000	W

GPIO Port 3 Data Out Clear Register

Address: 0x400200DC, Reset: 0x0000, Name: GP3CLR

Table 366. Bit Descriptions for GP3CLR

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	CLR	Set the output low for the port pin. Each bit is set to drive the corresponding GPIO pin low. Clearing this bit has no effect.	0x000	W

GPIO Port 3 Pin Toggle Register

Address: 0x400200E0, Reset: 0x0000, Name: GP3TGL

Table 367. Bit Descriptions for GP3TGL

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	TGL	Toggle the output of the port pin. Each bit is set to invert the corresponding GPIO pin. Clearing this bit has not effect.	0x000	W

GPIO Port 3 Interrupt Polarity Register

Address: 0x400200E4, Reset: 0x0000, Name: GP3POL

Table 368. Bit Descriptions for GP3POL

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	INTPOL	Interrupt polarity. Determines whether interrupts are generated on the rising or falling edge of the corresponding GPIO pin. When cleared, an interrupt event is latched on a high to low transition. When set, an interrupt event is latched on a low to high transition.	0x000	RW

GPIO Port 3 Interrupt A Enable Register

Address: 0x400200E8, Reset: 0x0000, Name: GP3IENA

Table 369. Bit Descriptions for GP3IENA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	INTAEN	Interrupt A enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt A) or be masked. In either case, the occurrence of the event is captured in the GP3INT status register. When cleared, Interrupt A is not enabled (masked). When set, Interrupt A is enabled.	0x000	RW

GPIO Port 3 Interrupt B Enable Register

Address: 0x400200EC, Reset: 0x0000, Name: GP3IENB

Table 370. Bit Descriptions for GP3IENB

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	INTBEN	Interrupt B enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt B) or be masked. In either case, the occurrence of the event is captured in the GP3INT status register. When cleared, Interrupt B is not enabled (masked). When set, Interrupt B is enabled.	0x000	RW

GPIO Port 3 Interrupt Status Register

Address: 0x400200F0, Reset: 0x0000, Name: GP3INT

Table 371. Bit Descriptions for GP3INT

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
[14:0]	INTSTATUS	Interrupt status. Indicates that the appropriately configured rising or falling edge has been detected on the corresponding GPIO pin. After an event is detected, the INTSTATUS bit remains set until cleared. This is true even if the GPIO pin transitions back to a nonactive state. INTSTATUS bits are cleared by writing 1 to the appropriate bit location. Writing 0 has no effect.	0x000	RW1C

GPIO Port 4 Configuration Register

Address: 0x40020100, Reset: 0x00000000, Name: GP4CON

The GP4CON register is reserved for top level pin muxing for the GPIO block.

Note that any combination not documented are reserved and must not be used.

Table 372. Bit Descriptions for GP4CON

Bits	Bit Name	Description	Reset	Access
[31:6]	RESERVED	Reserved.	0x0	R
[5:4]	PIN2_CFG	P4.2 configuration bits. 00: GPIO. 01: Timer B PWM output.	0x0	RW
[3:2]	PIN1_CFG	P4.1 configuration bits. 00: GPIO. 01: I^2C SDATA line.	0x0	RW
[1:0]	PIN0_CFG	P4.0 configuration bits. 00: GPIO. 01: I^2C SCLK line.	0x0	RW

GPIO Port 4 Output Enable Register

Address: 0x40020104, Reset: 0x0000, Name: GP4OEN

Table 373. Bit Descriptions for GP4OEN

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	OEN	Pin output drive enable. Each bit is set to enable the output for that particular pin. It is cleared to disable the output for each pin.	0x0	RW

GPIO Port 4 Output Pull-Up/Pull-Down Enable Register

Address: 0x40020108, Reset: 0x0000, Name: GP4PE

Table 374. Bit Descriptions for GP4PE

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	PE	Pin pull enable. Each bit is set to enable the pull-up/pull-down for that particular pin. It is cleared to disable the pull-up/pull-down for each pin.	0x0	RW

GPIO Port 4 Input Path Enable Register

Address: 0x4002010C, Reset: 0x0000, Name: GP4IEN

Table 375. Bit Descriptions for GP4IEN

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	IEN	Input path enable. Each bit is set to enable the input path and cleared to disable the input path for the GPIO pin.	0x0	RW

GPIO Port 4 Registered Data Input

Address: 0x40020110, Reset: 0x0000, Name: GP4IN

Table 376. Bit Descriptions for GP4IN

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	IN	Registered data input. Each bit reflects the state of the GPIO pin if the corresponding input buffer is enabled. If the pin input buffer is disabled the value seen is zero.	0x0	R

GPIO Port 4 Data Output Register

Address: 0x40020114, Reset: 0x0000, Name: GP4OUT

Table 377. Bit Descriptions for GP4OUT

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	OUT	Data out. Set by user code to drive the corresponding GPIO high. Cleared by user to drive the corresponding GPIO low.	0x0	RW

GPIO Port 4 Data Out Set Register

Address: 0x40020118, Reset: 0x0000, Name: GP4SET

Table 378. Bit Descriptions for GP4SET

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	SET	Set the output high for the pin. Set by user code to drive the corresponding GPIO high. Clearing this bit has no effect.	0x0	W

GPIO Port 4 Data Out Clear Register

Address: 0x4002011C, Reset: 0x0000, Name: GP4CLR

Table 379. Bit Descriptions for GP4CLR

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	CLR	Set the output low for the port pin. Each bit is set to drive the corresponding GPIO pin low. Clearing this bit has no effect.	0x0	W

GPIO Port 4 Pin Toggle Register

Address: 0x40020120, Reset: 0x0000, Name: GP4TGL

Table 380. Bit Descriptions for GP4TGL

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	TGL	Toggle the output of the port pin. Each bit is set to invert the corresponding GPIO pin. Clearing this bit has not effect.	0x0	W

GPIO Port 4 Interrupt Polarity Register

Address: 0x40020124, Reset: 0x0000, Name: GP4POL

Table 381. Bit Descriptions for GP4POL

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	INTPOL	Interrupt polarity. Determines whether interrupts are generated on the rising or falling edge of the corresponding GPIO pin. When cleared, an interrupt event is latched on a high to low transition. When set, an interrupt event is latched on a low to high transition.	0x0	RW

GPIO Port 4 Interrupt A Enable Register

Address: 0x40020128, Reset: 0x0000, Name: GP4IENA

Table 382. Bit Descriptions for GP4IENA

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	INTAEN	Interrupt A enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt A) or be masked. In either case, the occurrence of the event is captured in the GP4INT status register. When cleared, Interrupt A is not enabled (masked). When set Interrupt A, is enabled.	0x0	RW

GPIO Port 4 Interrupt B Enable Register

Address: 0x4002012C, Reset: 0x0000, Name: GP4IENB

Table 383. Bit Descriptions for GP4IENB

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	INTBEN	Interrupt B enable. Determines if a latched edge event is allowed to interrupt the core (Interrupt B) or be masked. In either case, the occurrence of the event is captured in the GP4INT status register. When set, Interrupt B is not enabled (masked). When cleared, Interrupt B is enabled.	0x0	RW

GPIO Port 4 Interrupt Status Register

Address: 0x40020130, Reset: 0x0000, Name: GP4INT

Table 384. Bit Descriptions for GP4INT

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:0]	INTSTATUS	Interrupt status. Indicates that the appropriately configured rising or falling edge has been detected on the corresponding GPIO pin. After an event is detected, the INTSTATUS bit remains set until cleared. This is true even if the GPIO pin transitions back to a nonactive state. INTSTATUS bits are cleared by writing 1 to the appropriate bit location. Writing 0 has no effect.	0x0	RW1C

I²C SERIAL INTERFACE

FUNCTIONALITY

This I²C interface provides both master and slave functionality. The peripheral complies with the I²C bus specification, Version 2.1.

Features include the following:

- 2-byte transmit and receive FIFOs for the master and slave.
- Support for repeated starts.
- Support for 10-bit addressing.
- Master arbitration is supported.
- Continuous read mode for the master or up to 512 bytes fixed read.
- Support for four 7-bit device addresses in the slave or one 10-bit address and two 7-bit addresses.
- Support for internal and external loopback.
- Support for DMA.
- Support for bus clear.

The I²C bus peripheral has two pins used for data transfer. SCL is a serial clock, and SDA is a serial data pin. The pins are configured in a wired-AND format that allows arbitration in a multimaster system.

A master device can be configured to generate the serial clock. The frequency is programmed by the user in the serial clock divisor register. The master channel can be set to operate in fast mode (400 kHz) or standard mode (100 kHz).

The I²C bus peripherals address in the I²C bus system is programmed by the user. This ID can be changed at any time while a transfer is not in progress. The user can set up to four slave addresses to be recognized by the peripheral. The peripheral is implemented with a 2-byte FIFO for each transmit and receive shift register. IRQ pins and status bits in the control registers are available to signal to the processor core when the FIFOs must be serviced.

OPERATION

The GPIOs used for I²C communication must be configured in I²C mode before enabling the I²C peripheral.

Master Transfer Initiation

If the master enable bit (MASEN) is set, a master transfer sequence can be initiated by writing a value to the I2CADRx register. If there is valid data in the I2CxTX register, it is the first byte transferred in the sequence after the address byte during a write sequence.

Slave Transfer Initiation

If the slave enable bit (SLVEN) is set, a slave transfer sequence is monitored for the device address in the I2CID0, I2CID1, I2CID2, or I2CID3 registers. If the device address is recognized, the device participates in the slave transfer sequence, as described in Figure 72 and Figure 73. Note that a slave operation always starts with the assertion of one of three interrupt sources: MRXREQ/SRXREQ, MTXREQ/STXREQ, or GCINT. The software must always look for a stop interrupt to ensure that the transaction has completed correctly and to deassert the stop interrupt status bit.

Rx/Tx Data FIFOs

The transmit datapath for both master and slave consists of Tx FIFOs that are two bytes deep, MTX and STX, and a transmit shifter. The transmit status bits, Bits[1:0] in the I2CMSTA register and Bit 0 in the I2CSSTA register, denote whether there is valid data in the Tx FIFO. Data from the Tx FIFO is loaded into the Tx shifter when a serial byte begins transmission. If the Tx FIFO is not full during an active transfer sequence, the transmit request bit (MTXREQ/STXREQ) in I2CMSTA or I2CSSTA asserts.

In the slave, if there is no valid data to transmit when the Tx shifter is loaded, the transmit underflow status bit asserts.

The master generates a stop condition if there is no data in the transmit FIFO and the master is writing data (direction bit = 0).

The receive datapath consists of a master and slave Rx FIFO, which are each two bytes deep: I2CMRX and I2CSRX. The receive request interrupt bits (MRXREQ/SRXREQ) in I2CMSTA or I2CSSTA indicate if there is valid data in the Rx FIFO. Data is loaded into the Rx FIFO after each byte is received.

If valid data in the Rx FIFO is overwritten by the Rx shifter, the receive overflow status bit (MRXOF/SRXOF) in I2CMSTA or I2CSSTA asserts.

If the Tx FIFO of the slave is loaded with a byte whose MSB is 0 just on the rising edge of SCL for the acknowledge/no acknowledge, the slave pulls the SDA low and holds the line until the device is reset. To avoid this process, Make sure the Tx FIFO is always loaded on time by preloading Tx FIFO in the preceding Rx interrupt.

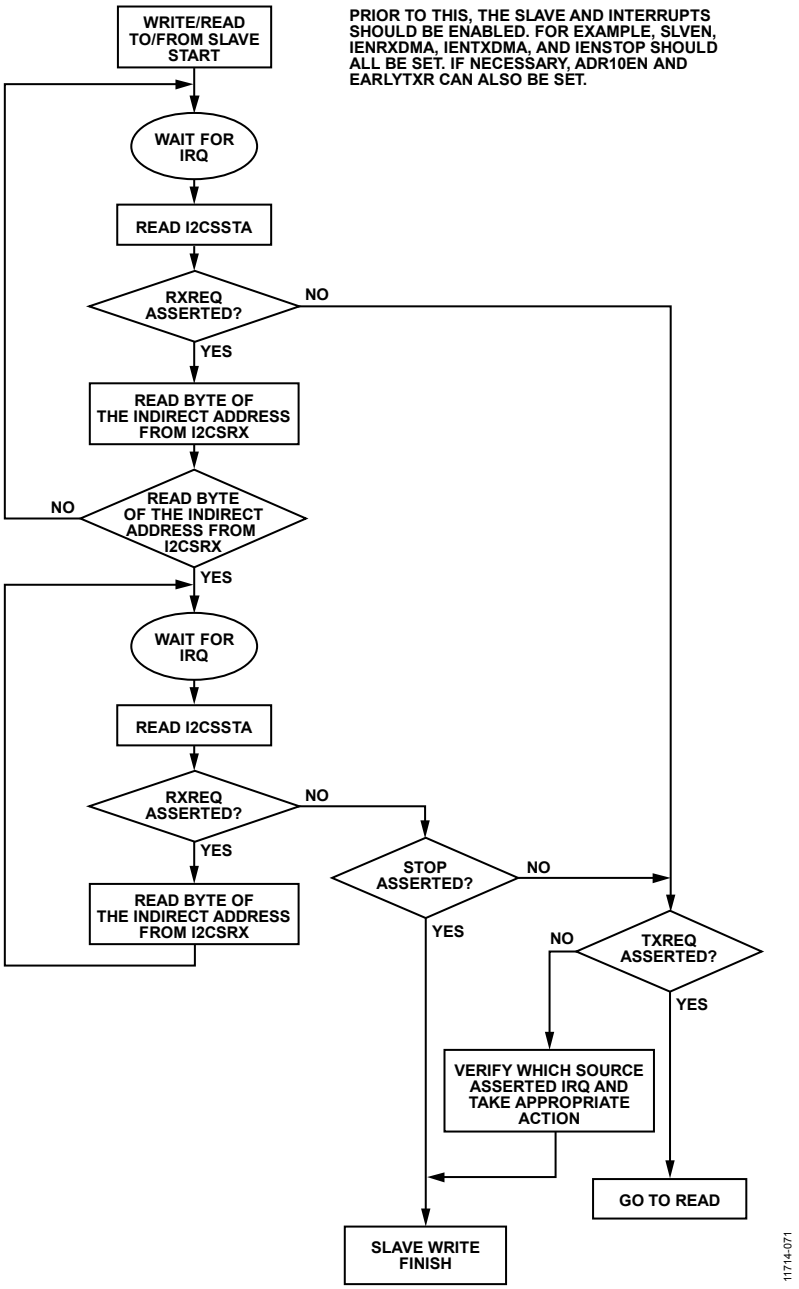


Figure 72. Slave Read/Write Flow

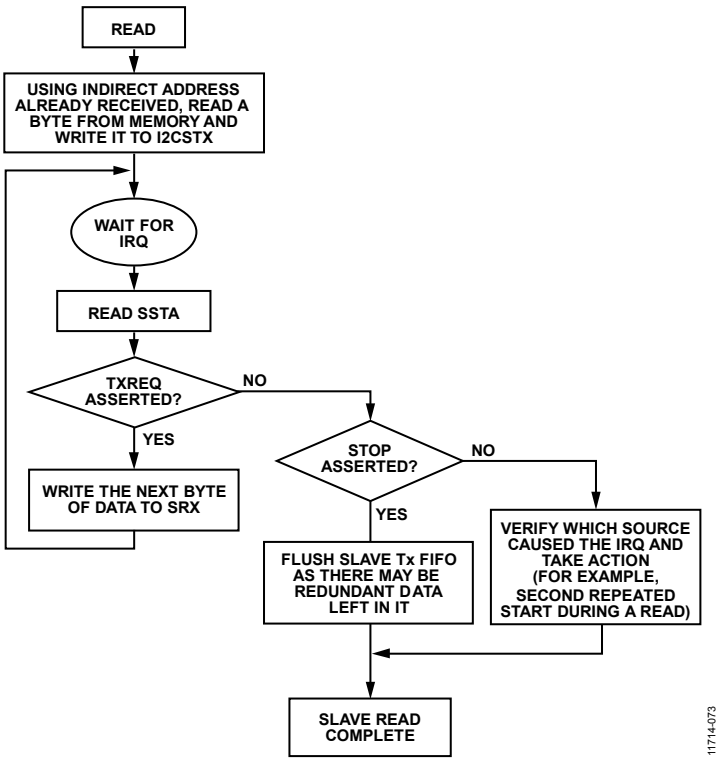


Figure 73. Slave Read/Write Flow Continued: Read Portion

Master No Acknowledge

When receiving data, the master responds with a no acknowledge if its FIFO is full and an attempt is made to write another byte to the FIFO. This last byte received is not written to the FIFO and is lost.

No Acknowledge from the Slave

If the slave does not acknowledge a read access, simply not writing data into the slave transmit FIFO results in a no acknowledge.

If the slave does not acknowledge a master write, assert the NACK bit in the slave control register.

Normally, the slave acknowledges all bytes written into the receive FIFO. If the receive FIFO fills up, the slave cannot write further bytes to it, and it does not acknowledge subsequent bytes not written to the FIFO. The master must then stop the transaction.

The slave does not acknowledge a matching device address if the direction bit is 1 (read) and the transmit FIFO is empty. Therefore, there is very little time for the microcontroller to respond to a slave transmit request and the assertion of acknowledge. It is recommended that EARLYTXR be asserted for this reason.

General Call

If the general call enable bit (GCEN) and the slave enable bit (SLVEN) are set, the device can respond to a general call. If the second byte of the general call is 0x06, the I^2C interface (master and slave) is reset. The general call interrupt status asserts, and the general call ID bits (GCID) are 0x1. User code must take the correct action; that is, user code must reset the entire system or reenable the I^2C interface.

If the second byte is 0x04 (write-programmable part of slave address by hardware), the general call interrupt status bit is asserted, and the general call ID (GCID) is 0x2.

The general call interrupt status bit is set on any general call after the second byte is received, and user code must take the correct action, including reprogramming the device address.

If GCEN is asserted, the slave always acknowledged the first byte of a general call. It also acknowledges the second byte of a general call if the second byte is 0x04 or 0x06, or if the second byte is a hardware general call and HGCEN is asserted.

The ALT register contains the alternate device ID for a hardware general call sequence. If the hardware general call enable bit (HGCEN) and the GCEN and SLVEN bits are set, the device can recognize a hardware general call. When a general call sequence is issued and the second byte of the sequence is identical to ALT, the hardware call sequence is recognized for the device.

Generation of Repeated Starts by the Master

The master generates a repeated start if the first master address byte register is written when the master is busy performing a transaction. After the state machine starts to transmit the device address, the user can write to the first master address byte register.

For example, if a write repeated start read/write transaction is required, write to the first master address byte register after the state machine starts to transmit the device address or after the first TXREQ interrupt is received. When the transmit FIFO empties, a repeated start is generated.

Similarly, if a read repeated start read/write transaction is required, write to the first master address byte register after the state machine starts to transmit the device address, or after the first RXREQ interrupt is received. When the requested receive count is reached, a repeated start is generated.

DMA Requests

Three DMA channels are required to service the I²C master and slave. DMA enable bits are provided in the slave control register and in the master control register.

I²C Reset Mode

The slave state machine is reset when SLVEN is written to 0, and the master state machine is reset when MASEN is written to 0.

I²C Test Modes

The device can be placed in an internal loopback mode by setting the LOOPBACK bit in the I2CMCON register. There are four FIFOs (master Tx, master Rx, slave Tx, and slave Rx); therefore, the I²C peripheral can in effect be set up to talk to itself. External loopback can be performed if the master is set up to address the address of the slave.

I²C Low Power Mode

If the master and slave are both disabled (MASEN = SLVEN = 0), the device is in its lowest power mode.

I²C Bus Clear Operation

If the master lost arbitration when the BUS_CLR_EN bit is set, the master sends out an extra nine SCL cycles so that the slave holding the SDA line can release it any time before those nine SCL cycles elapse. If the PRESTOP_BUS_CLR bit is asserted, the master stops sending the SCL clocks after SDA is released.

Power-Down Considerations

The following must be considered when the device is being powered down to hibernate mode.

If the master/slave is idle (which can be determined using the respective status registers), it can be immediately disabled by clearing the MASEN/SLVEN bits in the master/slave control registers, respectively.

If master/slave is active, there are four possible scenarios.

- I²C is a master performing Rx: In this case, the device is receiving data based on the count programmed in the I2CMRXCNT register. It is in continuous read mode if the EXTEND bit (I2CMRXCNT[8]) is set. To stop the read transfer, clear the EXTEND bit and assign the I2CMRXCNT register with I2CMCRXCNT + 1, where I2CMCRXCNT is the current read count and +1 signifies that there is some room for the completion of the transaction. If the newly programmed value is less than the current count, data is received until the current count reaches the programmed count. This ends the transfer after receiving the next byte. After the transaction complete interrupt is received, the core must disable the master by clearing the MASEN bit.
- I²C is a master performing Tx: The software flushes the Tx FIFO by setting MFLUSH (Bit 9 of the I2CFSTA register) and disables the Tx request by clearing IENMTX (Bit 5 of the I2CMCON register). This ends the current transfer after transmitting the byte in progress. When the transaction complete interrupt is received, the core must clear the MASEN bit in the I2CMCON register. Disabling the master before completion can cause the bus to hang indefinitely.
- I²C is a slave performing Rx: The software sets the NACK bit (Bit 7 of the I2CSCON register). This results in a no acknowledge for the next communication, after which the external master must stop. Upon receiving the stop interrupt, the core disables the slave by clearing the SLVEN bit of the I2CSCON register.
- I²C is a slave performing Tx: After a slave transmit begins, it cannot not acknowledge any subsequent transaction (acknowledge is driven only by the master). Therefore, it must wait until the external master issues a stop condition. After receiving the stop interrupt, the slave can be disabled. This is a clean way to exit. However, if the slave must be disabled immediately, this can be done only at the cost of wrong data being transmitted (all FFs). This is because the SDA line is driven any more, and it is pulled up during data phase. Note that the bus does not hang in this case.

I²C MEMORY MAPPED REGISTERS

I²C Register Map

Table 385. I2CMS Register Summary

Address	Name	Description	Reset	RW
0x40003000	I2CMCON	Master control	0x0000	RW
0x40003004	I2CMSTA	Master status	0x6000	R
0x40003008	I2CMRX	Master receive data	0x0000	R
0x4000300C	I2CMTX	Master transmit data	0x0000	RW
0x40003010	I2CMRXCNT	Master receive data count	0x0000	RW
0x40003014	I2CMCRXCNT	Master current receive data count	0x0000	R
0x40003018	I2CADR1	First master address byte	0x0000	RW
0x4000301C	I2CADR2	Second master address byte	0x0000	RW
0x40003020	I2CBYT	Start byte	0x0000	RW
0x40003024	I2CDIV	Serial clock period divisor	0x1F1F	RW
0x40003028	I2CSCON	Slave control	0x0000	RW
0x4000302C	I2CSSTA	Slave I²C status/error/IRQ	0x0001	R
0x40003030	I2CSRX	Slave receive	0x0000	R
0x40003034	I2CSTX	Slave transmit	0x0000	RW
0x40003038	I2CALT	Hardware general call ID	0x0000	RW
0x4000303C	I2CID0	First slave address device ID	0x0000	RW
0x40003040	I2CID1	Second slave address device ID	0x0000	RW
0x40003044	I2CID2	Third slave address device ID	0x0000	RW
0x40003048	I2CID3	Fourth slave address device ID	0x0000	RW
0x4000304C	I2CFSTA	Master and slave FIFO status	0x0000	RW
0x40003050	I2CSHCON	Shared control	0x0000	W
0x40003054	I2CTCTL	Timing control register	0x0005	RW

Master Control Register

Address: 0x40003000, Reset: 0x0000, Name: I2CMCON

Table 386. Bit Descriptions for I2CMCON

Bits	Bit Name	Description	Reset	Access
[15:14]	Reserved	Reserved.	0x0	R
13	PRESTOP_BUS_CLR	Prestop bus clear. This bit is used in conjunction with BUS_CLR_EN. If this bit is set, the master stops sending any more SCL clocks if SDA is released before nine SCL cycles.	0x0	RW
12	BUS_CLR_EN	Bus clear enable. If this bit is set, the master initiates a bus clear operation by sending up to nine extra SCL cycles if the arbitration was lost. This bit is added to come out of a SDA stuck situation due to a misbehaving slave and, thus, it must be used with caution. It must be set only when there is no other active I²C master.	0x0	RW
11	MTXDMA	Enable master Tx DMA request. Set to 1 by user code to enable I²C master DMA Tx requests. Cleared by user code to disable DMA mode.	0x0	W
10	MRXDMA	Enable master Rx DMA request. Set to 1 by user code to enable I²C master DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	W
9	MXMITDEC	Decrement master Tx FIFO status when a byte has been transmitted. If set to 1, the master transmit FIFO status is decremented when a byte has been transmitted. If set to 0, the master transmit FIFO status is decremented when the byte is unloaded from the FIFO into a shadow register at the start of byte transmission.	0x0	RW
8	IENCMP	Transaction completed (or stop detected) interrupt enable. Transaction completed interrupt enable. When asserted, an interrupt is generated when a STOP is detected. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
7	IENACK	Acknowledge not received interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
6	IENALOST	Arbitration lost interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
5	IENMTX	Transmit request interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
4	IENMRX	Receive request interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
3	Reserved	Reserved.	0x0	RW
2	LOOPBACK	Internal loopback enable. When this bit is set SCL and SDA out of the device are muxed onto their corresponding inputs. Note that is also possible for the master to loop back a transfer to the slave as long as the device address corresponds, that is, external loopback.	0x0	RW
1	COMPLETE	Start back-off disable. Setting this bit enables the device to compete for ownership even if another device is currently driving a START condition.	0x0	RW
0	MASEN	Master enable. When the bit is 1 the master is enabled. When the bit is 0 all master state machine flops are held in reset and the master is disabled. The master must be disabled when not in use as this gates the clock to the master and save power. This bit must not be cleared until a transaction has completed, see the TCOMP bit in the master status register. Note that APB writable register bits are not reset by this bit. Cleared by default.	0x0	RW

Master Status Register

Address: 0x40003004, Reset: 0x6000, Name: I2CMSTA

Table 387. Bit Descriptions for I2CMSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	SCL_FILTERED	State of SCL Line. This bit is the output of the glitch-filter on SCL. SCL is always pulled high when undriven.	0x1	R
13	SDA_FILTERED	State of SDA Line. This bit is the output of the glitch-filter on SDA. SDA is always pulled high when undriven.	0x1	R
12	MTXUFLOW	Master Transmit Underflow. Asserts when the I^2C master ends the transaction due to Tx-FIFO empty condition. This bit is asserted only when the IENMTX bit is set.	0x0	RC
11	MSTOP	STOP driven by this I^2C Master. Asserts when this I^2C master drives a STOP condition on the I^2C bus. This bit, when asserted, can indicate a Transaction completion, Tx-underflow, Rx-overflow or a no acknowledge by the slave. This is different from the TCOMP as this bit is not asserted when the STOP condition occurs due to any other I^2C master. No interrupt is generated for the assertion of this bit. However, if IENCMP is 1, every STOP condition generates an interrupt and this bit can be read. When this bit is read, it clears status.	0x0	RC
10	LINEBUSY	Line is busy. Asserts when a START is detected on the I^2C bus. De-asserts when a STOP is detected on the I^2C bus.	0x0	R
9	MRXOF	Master receive FIFO overflow. Asserts when a byte is written to the receive FIFO when the FIFO is already full. When the bit is read, it clears status.	0x0	RC
8	TCOMP	Transaction complete or stop detected. Transaction complete. This bit asserts when a STOP condition is detected on the I^2C bus. If IENCMP is 1, an interrupt is generated when this bit asserts. This bit only asserts if the master is enabled (MASEN = 1). This bit must be used to determine when it is safe to disable the master. It can also be used to wait for another master transaction to complete on the I^2C bus when this master loses arbitration. When this bit is read, it clears the status. This bit can drive an interrupt.	0x0	RC

Bits	Bit Name	Description	Reset	Access
7	NACKDATA	Acknowledge not received in response to data write. This bit asserts when an acknowledge is not received in response to a data write transfer. If IENACK is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt. This bit is cleared on a read of the I2CMSTA register.	0x0	RC
6	MBUSY	Master busy. This bit indicates that the master state machine is servicing a transaction. It is cleared if the state machine is idle or another device has control of the I^2C bus.	0x0	R
5	ALOST	Arbitration lost. This bit asserts if the master loses arbitration. If IENALOST is 1, an interrupt is generated when this bit asserts. This bit is cleared on a read of the I2CMSTA register. This bit can drive an interrupt.	0x0	RC
4	NACKADDR	Acknowledge not received in response to an address. This bit asserts if an acknowledge is not received in response to an address. If IENACK is 1, an interrupt is generated when this bit asserts. This bit is cleared on a read of the I2CMSTA register. This bit can drive an interrupt.	0x0	RC
3	MRXREQ	Master receive request. This bit asserts when there is data in the receive FIFO. If IENMRX is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt.	0x0	R
2	MTXREQ	Master transmit request. This bit asserts when the direction bit is 0 and the transmit FIFO is either empty or not full. If IENMTX is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt.	0x0	R
[1:0]	MTXFSTA	Master transmit FIFO status. These two bits show the master transmit FIFO status and can be decoded as follows: 00: FIFO empty. 10: 1 byte in FIFO. 11: FIFO full. 01: not used.	0x0	R

Master Receive Data Register

Address: 0x40003008, Reset: 0x0000, Name: I2CMRX

Table 388. Bit Descriptions for I2CMRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	I2CMRX	Master receive register. This register allows access to the receive data FIFO. The FIFO can hold two bytes.	0x0	R

Master Transmit Data Register

Address: 0x4000300C, Reset: 0x0000, Name: I2CMTX

Table 389. Bit Descriptions for I2CMTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	I2CMTX	Master transmit register. For test and debug purposes, when read, this register returns the byte that is currently being transmitted by the master. That is, a byte written to the transmit register can be read back some time later when that byte is being transmitted on the line. This register allows access to the transmit data FIFO. The FIFO can hold two bytes.	0x0	RW

Master Receive Data Count Register

Address: 0x40003010, Reset: 0x0000, Name: I2CMRXCNT

Table 390. Bit Descriptions for I2CMRXCNT

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	EXTEND	Extended read. Use this bit if greater than 256 bytes are required on a read. For example, to receive 412 bytes write 0x100 (EXTEND = 1) to the I2CMRXCNT register. Wait for the first byte to be received, then check the I2CMCRXCNT register for every byte received thereafter. When COUNT returns to 0, 256 bytes have been received. Then write 0x09C to the I2CMRXCNT register.	0x0	RW
[7:0]	COUNT	Receive count. Program the number of bytes required minus one to this register. If just 1 byte is required write 0 to this register. If greater than 256 bytes are required, use EXTEND.	0x0	RW

Master Current Receive Data Count Register

Address: 0x40003014, Reset: 0x0000, Name: I2CMCRXCNT

Table 391. Bit Descriptions for I2CMCRXCNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	COUNT	Current receive count. This register gives the total number of bytes received so far. If 256 bytes are requested, this register reads 0 when the transaction has completed.	0x0	R

First Master Address Byte Register

Address: 0x40003018, Reset: 0x0000, Name: I2CADR1

Table 392. Bit Descriptions for I2CADR1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ADR1	Address Byte 1. If a 7-bit address is required, Bits[7:1] of ADR1 are programmed with the address and Bit 0 of ADR1 is programmed with the direction (read or write). If a 10-bit address is required, Bits[7:3] of ADR1 are programmed with 11110, Bits[2:1] of ADR1 are programmed with the two MSBs of the address, and Bit 0 of ADR1 is programmed to 0.	0x0	RW

Second Master Address Byte Register

Address: 0x4000301C, Reset: 0x0000, Name: I2CADR2

Table 393. Bit Descriptions for I2CADR2

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ADR2	Address Byte 2. This register is only required when addressing a slave with a 10-bit address. Bits[7:0] of ADR2 are programmed with the lower eight bits of the address.	0x0	RW

Start Byte Register

Address: 0x40003020, Reset: 0x0000, Name: I2CBYT

Table 394. Bit Descriptions for I2CBYT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	SBYTE	Start byte. This register can be used to generate a start byte at the start of a transaction. To generate a start byte followed by a normal address, first write to VALUE then write to the address register (I2CADR1). This drives the byte written in VALUE on to the bus followed by a repeated start. This register can be used to drive any byte on to the I^2C bus followed by a repeated start (not just a start byte 00000001).	0x0	RW

Serial Clock Period Divisor Register

Address: 0x40003024, Reset: 0x1F1F, Name: I2CDIV

Table 395. Bit Descriptions for I2CDIV

Bits	Bit Name	Description	Reset	Access
[15:8]	HIGH	Serial clock high time. This register controls the clock high time. The timer is driven by the core clock (UCLK). Use the following equation to derive the required high time: HIGH = (REQD_HIGH_TIME/UCLK_PERIOD) − 2. For example, to generate a 400 kHz SCL with a low time of 1300 ns and a high time of 1200 ns, with a core clock frequency of 50 MHz, LOTIME = 1300 ns/20 ns − 1 = 0x40 (64 decimal) and HIGH = 1200 ns/20 ns − 2 = 0x3A (58 decimal). This register is reset to 0x1F which gives an SCL high time of 33 UCLK ticks. t_{SHD} is also determined by the HIGH. t_{SHD} = (HIGH − 1) × UCLK_PERIOD. Because t_{SHD} must be 600 ns, with UCLK = 50 MHz, the minimum value for HIGH is 31. This gives an SCL high time of 660 ns.	0x1F	RW
[7:0]	LOW	Serial clock low time. This register controls the clock low time. The timer is driven by the core clock (UCLK). Use the following equation to derive the required low time: LOW = (REQD_LOW_TIME/UCLK_PERIOD) − 1. This register is reset to 0x1F, which gives an SCL low time of 32 UCLK ticks.	0x1F	RW

Slave Control Register

Address: 0x40003028, Reset: 0x0000, Name: I2CSCON

Table 396. Bit Descriptions for I2CSCON

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	STXDMA	Enable slave Tx DMA request. Set to 1 by user code to enable I^2C slave DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	RW
13	SRXDMA	Enable slave Rx DMA request. Set to 1 by user code to enable I^2C slave DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	RW
12	IENREPST	Repeated start interrupt enable. If set to 1, an interrupt is generated when the REPSTART status bit asserts. If set to 0, an interrupt is not generated when the REPSTART status bit asserts.	0x0	RW
11	SXMITDEC	Decrement slave Tx FIFO status when a byte has been transmitted. If set to 1, the transmit FIFO status is decremented when a byte has been transmitted. If set to 0, the transmit FIFO status is decremented when the byte is unloaded from the FIFO into a shadow register at the start of byte transmission.	0x0	RW
10	IENSTX	Slave transmit request interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
9	IENSRX	Slave receive request interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
8	IENSTOP	Stop condition detected interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
7	NACK	No acknowledge next communication. If this bit is set, the next communication is not acknowledged. This can be used, for example, if during a 24xx style access, an attempt was made to write to a read only or nonexisting location in system memory. That is the indirect address in a 24xx style write pointed to an unwritable memory location.	0x0	RW
6	RESERVED	Reserved.	0x0	RW
5	EARLYTXR	Early transmit request mode. Setting this bit enables a transmit request just after the positive edge of the direction bit SCL clock pulse.	0x0	RW
4	GCSBCLR	General call status bit clear. The general call status and general call ID bits are cleared when a 1 is written to this bit. The general call status and General Call ID bits are not reset by anything other than a write to this bit or a full reset.	0x0	W

Bits	Bit Name	Description	Reset	Access
3	HGCEN	Hardware general call enable. When this bit and the general call enable bit are set, the device after receiving a general call, Address 00H and a data byte checks the contents of ALT against the receive shift register. If they match, the device has received a hardware general call. This is used if a device needs urgent attention from a master device without knowing which master it needs to turn to. This is a call to whom it may concern. The device that requires attention embeds its own address into the message. The LSB of the ALT bit of the I2CALT register must always be written to with a 1, as per the I²C January 2000 specification.	0x0	RW
2	GCEN	General call enable. This bit enables the I²C slave to acknowledge an I²C general call, Address 0x00 (write).	0x0	RW
1	ADR10EN	Enabled 10-bit addressing. If this bit is clear, the slave can support four slave addresses, programmed in Register I2CID0 to Register I2CID3. When this bit is set, 10-bit addressing is enabled. One 10-bit address is supported by the slave and is stored in I2CID0 and I2CID1, where I2CID0 contains the first byte of the address and the upper 5 bits must be programmed to 11110. I2CID3 and I2CID4 can be programmed with 7-bit addresses at the same time.	0x0	RW
0	SLVEN	Slave enable. When set to 1, the slave is enabled. When set to 0, all slave state machine flops are held in reset and the slave is disabled. Note that APB writable register bits are not reset.	0x0	RW

Slave I²C Status/Error/IRQ Register

Address: 0x4000302C, Reset: 0x0001, Name: I2CSSTA

Table 397. Bit Descriptions for I2CSSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED		0x0	R
14	START	Start and matching address. This bit is asserted if a start is detected on SCL/SDA and the device address matched, or a general call, GC, (Address = 0000_0000) code is received and GC is enabled, or a high speed, HS, (Address = 0000_1XXX) code is received, or a start byte (0000_0001) is received. It is cleared on receipt of either a stop or start condition.	0x0	R
13	REPSTART	Repeated start and matching address. This bit is asserted if START is already asserted and then a repeated start is detected. It is cleared when read or on receipt of a STOP condition. This bit can drive an interrupt.	0x0	RC
[12:11]	IDMAT	Device ID matched. 00: received address matched ID Register 0. 01: received address matched ID Register 1. 10: received address matched ID Register 2. 11: received address matched ID Register 3.	0x0	R
10	STOP	Stop after start and matching address. This bit is set by hardware if the slave device received a STOP condition after a previous START condition and a matching address. Cleared by a read of the status register. If IENSTOP in the slave control register is asserted, the slave interrupt request asserts when this bit is set. This bit can drive an interrupt.	0x0	RC
[9:8]	GCID	General ID. GCID is cleared when the GCSBCLR is written to 1. These status bits are not cleared by a general call reset. 00: no general call. 01: general call reset and program address. 10: general call program address. 11: general call matching alternative ID.	0x0	R
7	GCINT	General call interrupt. This bit always drives an interrupt. The bit is asserted if the slave device receives a general call of any type. To clear, write 1 to the GCSBCLR in the slave control register. If it was a general call reset, all registers are at their default values. If it was a hardware general call, the Rx FIFO holds the second byte of the general call, and this can be compared with the ALT register.	0x0	R
6	SBUSY	Slave busy. Set by hardware if the slave device receives an I²C START condition. Cleared by hardware when the address does not match an ID register, the slave device receives a I²C STOP condition, or if a repeated start address does not match.	0x0	R
5	NOACK	Acknowledge not generated by the slave. When asserted, it indicates that the slave responded to its device address with a NOACK. It is asserted if there was no data to transmit and sequence was a slave read or if the NACK bit was set in the slave control register and the device was addressed. This bit is cleared on a read of the I2CSSTA register.	0x0	RC
4	SRXOF	Slave receive FIFO overflow. Asserts when a byte is written to the slave receive FIFO when the FIFO is already full.	0x0	RC

Bits	Bit Name	Description	Reset	Access
3	SRXREQ	Slave receive request. SRXREQ asserts whenever the slave receive FIFO is not empty. Read or flush the slave receive FIFO to clear this bit. This bit asserts on the falling edge of the SCL clock pulse that clocks in the last data bit of a byte. This bit can drive an interrupt.	0x0	RC
2	STXREQ	Slave transmit request. If EARLYTXR = 0, STXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted on the negative edge of the SCL pulse that clocks in the direction bit (if the device address matched also). If EARLYTXR = 1, STXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted after the positive edge of the SCL pulse that clocks in the direction bit (if the device address matched also). This bit is cleared on a read of the I2CSSTA register.	0x0	RC
1	STXUR	Slave transmit FIFO underflow. Set to 1 if a master requests data from the device and the Tx FIFO is empty for the rising edge of SCL. Set to 0 if no such FIFO underflow detected.	0x0	RC
0	STXFSEREQ	Slave Tx FIFO status or early request. If EARLYTXR = 0, this bit is asserted whenever the slave Tx FIFO is empty. If EARLYTXR = 1, STXFSEREQ is set when the direction bit for a transfer is received high. It asserts on the positive edge of the SCL clock pulse that clocks in the direction bit (if the device address matched also). It only asserts once for a transfer. It is cleared when read if EARLYTXR is asserted.	0x1	RW

Slave Receive Register

Address: 0x40003030, Reset: 0x0000, Name: I2CSRX

Table 398. Bit Descriptions for I2CSRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	I2CSRX	Slave receive register.	0x0	R

Slave Transmit Register

Address: 0x40003034, Reset: 0x0000, Name: I2CSTX

Table 399. Bit Descriptions for I2CSTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	I2CSTX	Slave transmit register.	0x0	RW

Hardware General Call ID Register

Address: 0x40003038, Reset: 0x0000, Name: I2CALT

Table 400. Bit Descriptions for I2CALT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ALT	Slave alternate. This register is used in conjunction with Bit 3 of I2CSCON to match a master generating a hardware general call. It is used in the case where a master device cannot be programmed with the address of a slave and instead the slave must recognize the address of the master.	0x0	RW

First Slave Address Device ID Register

Address: 0x4000303C, Reset: 0x0000, Name: I2CID0

Table 401. Bit Descriptions for I2CID0

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID0	Slave device ID 0. Bits[7:1] of I2CID0 is programmed with the device ID. Bit 0 of I2CID0 is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Second Slave Address Device ID Register

Address: 0x40003040, Reset: 0x0000, Name: I2CID1

Table 402. Bit Descriptions for I2CID1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID1	Slave device ID 1. Bits[7:1] of I2CID1 are programmed with the device ID. Bit 0 of I2CID1 is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Third Slave Address Device ID Register

Address: 0x40003044, Reset: 0x0000, Name: I2CID2

Table 403. Bit Descriptions for I2CID2

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID2	Slave device ID 2. Bits[7:1] of I2CID2 are programmed with the device ID. Bit 0 of I2CID2 is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Fourth Slave Address Device ID Register

Address: 0x40003048, Reset: 0x0000, Name: I2CID3

Table 404. Bit Descriptions for I2CID3

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID3	Slave device ID 3. Bits[7:1] of I2CID3 are programmed with the device ID. Bit 0 of I2CID3 is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Master and Slave FIFO Status Register

Address: 0x4000304C, Reset: 0x0000, Name: I2CFSTA

Table 405. Bit Descriptions for I2CFSTA

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	RW
9	MFLUSH	Flush the master transmit FIFO. Writing a 1 to this bit flushes the master transmit FIFO. The master transmit FIFO is flushed if arbitration is lost or a slave responds with a no acknowledge.	0x0	W
8	SFLUSH	Flush the slave transmit FIFO. Writing a 1 to this bit flushes the slave transmit FIFO.	0x0	W
[7:6]	MRXFSTA	Master receive FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: one byte in the FIFO. 10: two bytes in the FIFO. 11: reserved.	0x0	R
[5:4]	MTXFSTA	Master transmit FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: one byte in the FIFO. 10: two bytes in the FIFO. 11: reserved.	0x0	R
[3:2]	SRXFSTA	Slave receive FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: one byte in the FIFO. 10: two bytes in the FIFO. 11: reserved.	0x0	R

Bits	Bit Name	Description	Reset	Access
[1:0]	STXFSTA	Slave transmit FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: one byte in the FIFO. 10: two bytes in the FIFO. 11: reserved.	0x0	R

Shared Control Register

Address: 0x40003050, Reset: 0x0000, Name: I2CSHCON

Table 406. Bit Descriptions for I2CSHCON

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0	R
0	RESET	Reset START STOP detect circuit. Writing a 1 to this bit resets the SCL and SDA synchronizers, the START and STOP detect circuit, and the LINEBUSY detect circuit. These circuits are not reset when both the master and slave are disabled because LINEBUSY must assert even when the master is not enabled. It is only necessary to reset these circuits after a power on reset in case SCL/SDA do not power up cleanly. Reading this bit always reads back 0.	0x0	W

Timing Control Register

Address: 0x40003054, Reset: 0x0005, Name: I2CTCTL

Table 407. Bit Descriptions for I2CTCTL

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	FILTEROFF	Input filter control. It may be desirable to disable the digital filter if PCLK rate becomes <16 MHz. 0: digital filter is enabled and is equal to 1 PCLK. 1: digital filter is disabled. Filtering in the pad is unaffected.	0x0	RW
[7:5]	RESERVED	Reserved.	0x0	R
[4:0]	THDATIN	Data in hold start. THDATIN determines the hold time requirement that must be met before a start or stop condition is recognized. The hold time observed between SDA and SCL fall must exceed the programmed value to be recognized as a valid start. THDATIN = (t_{SHD})/(clock period). For example, at 16 MHz PCLK (62.5 ns) and setting a minimum t_{SHD} limit of 300 ns, (300 ns)/(62.5 ns) = 5.	0x5	RW

SERIAL PERIPHERAL INTERFACES (SPI)

FEATURES

SPI is an industry standard, synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, that is, full duplex. Transfers can operate to a maximum bit rate of 8 Mbps in both master and slave modes.

The SPI incorporates two DMA channels that interface with the DMA controller. One DMA channel is used for transmit and the other is used for receive.

The SPI features available include the following:

- Serial clock phase mode and serial clock polarity mode.
- LSB first transfer option.
- Loopback mode.
- Master or slave mode.
- Transfer and interrupt mode.
- Continuous transfer mode.
- Tx/Rx FIFO.
- Interrupt mode, interrupt after one, two, three, or four bytes.
- Rx overflow mode and Tx underflow mode.
- Wired-OR output mode.

OPERATION

The SPI port can be configured for master or slave operation and consists of four pins: MISO, MOSI, SCLK, and $\overline{\text{CS}}$. Note that the GPIOs used for SPI communication must be configured in SPI mode before enabling the SPI peripheral.

MISO (Master In, Slave Out) Pin

The MISO pin is configured as an input line in master mode and as an output line in slave mode. The MISO line on the master (data in) must be connected to the MISO line in the slave device (data out). The data is transferred as byte wide (8-bit) serial data, MSB first.

MOSI (Master Out, Slave In) Pin

The MOSI pin is configured as an output line in master mode and as an input line in slave mode. The MOSI line on the master (data out) must be connected to the MOSI line in the slave device (data in). The data is transferred as byte wide (8-bit) serial data, MSB first.

SCLK (Serial Clock I/O) Pin

The master serial clock (SCLK) synchronizes the data being transmitted and received through the MOSI SCLK period. Therefore, a byte is transmitted/received after eight SCLK periods. The SCLK pin is configured as an output in master mode and as an input in slave mode.

In master mode, the polarity and phase of the clock are controlled by the SPIxCON register, and the bit rate is defined in the SPIxDIV register as follows:

$$f_{SERIALCLOCK} = \frac{UCLK}{2 \times (1 + SPIxDIV)}$$

The maximum data rate is 8 Mbps.

In slave mode, the SPIxCON register must be configured with the phase and polarity of the expected input clock. The slave accepts data from an external master up to 8 Mbps.

In both master and slave modes, data is transmitted on one edge of the SCLK signal and sampled on the other. Therefore, it is important that the polarity and phase are configured the same for the master and slave devices.

Chip Select ($\overline{\text{CS}}$ Input) Pin

In SPI slave mode, a transfer is initiated by the assertion of the $\overline{\text{CS}}$ pin, which is an active low input signal. The SPI port then transmits and receives 8-bit data until the transfer is concluded by deassertion of the $\overline{\text{CS}}$ pin. In slave mode, the $\overline{\text{CS}}$ pin is always an input.

In SPI master mode, the $\overline{\text{CS}}$ pin is an active low output signal. It asserts itself automatically at the beginning of a transfer and deasserts itself upon completion.

SPI TRANSFER INITIATION

In master mode, the transfer and interrupt mode bit (TIM) determines the manner in which an SPI serial transfer is initiated. If the TIM bit is set, a serial transfer is initiated after a write to the Tx FIFO occurs. If the TIM bit is cleared, a serial transfer is initiated after a read of the Rx FIFO, and the read must be performed while the SPI interface is idle. A read performed during an active transfer does not initiate another transfer.

For any setting of the master mode enable (MASEN) and TIM bits, the SPI simultaneously receives and transmits data. Therefore, during data transmission, the SPI is receiving data and filling up the Rx FIFO. If the data is not read from the Rx FIFO, the overflow interrupt occurs after the FIFO overflows. If the user does not want to read the Rx data or receive overflow interrupts, the Rx FIFO flush enable (RFLUSH) bit can be set, and the receive data is not saved to the Rx FIFO. Similarly, when the user only wants to receive data and does not want to write data to the Tx FIFO, Bit 13 of the SPIxCON register (TFLUSH) can be set to avoid generating underflow interrupts from the Tx FIFO.

Tx Initiated Transfer

For transfers initiated by a write to the Tx FIFO, the SPI starts transmitting as soon as the first byte is written to the FIFO irrespective of the configuration in IRQ mode (MOD). The first byte is immediately read from the FIFO, written to the Tx shift register, and the transfer commences.

If the continuous transfer enable bit (CON) is set, the transfer continues until no valid data is available in the Tx FIFO. There are no stall periods between transfers if chip select is deasserted. The chip select asserts and remains asserted for the duration of the transfer until the Tx FIFO is empty. When the transfer stops does not depend on MOD; the transfer stops when there is no valid data left in the FIFO. Conversely, the transfer continues while there is valid data in the FIFO.

If CON is cleared, each transfer consists of a single 8-bit serial transfer. If valid data exists in the Tx FIFO, a new transfer is initiated after a stall period if chip select is deasserted.

Rx Initiated Transfer

Transfers initiated by a read of the Rx FIFO depend on the number of bytes to be received in the FIFO. If MOD is set to 11 and a read to the Rx FIFO occurs, the SPI initiates a 4-byte transfer. If continuous mode is set (CON), the four bytes are transferred continuously with no deassertion of chip select between bytes. If continuous mode is not set, the four bytes are transferred with a stall period between each transfer, and the chip select is deasserted. A read of the Rx FIFO while the SPI is receiving data does not initiate another transfer after the present transfer is complete.

In slave mode, a transfer is initiated by the assertion of the chip select of the device.

Note that only CS0 is active in slave mode.

The device as a slave transmits and receive 8-bit data until the transfer is concluded by the deassertion of chip select.

The SPI transfer protocol diagrams (Figure 74 and Figure 75) illustrate the data transfer protocol for the SPI and the effects of the CPHA and CPOL bits of the control register on that protocol, as indicated by T1, T2, and T3.

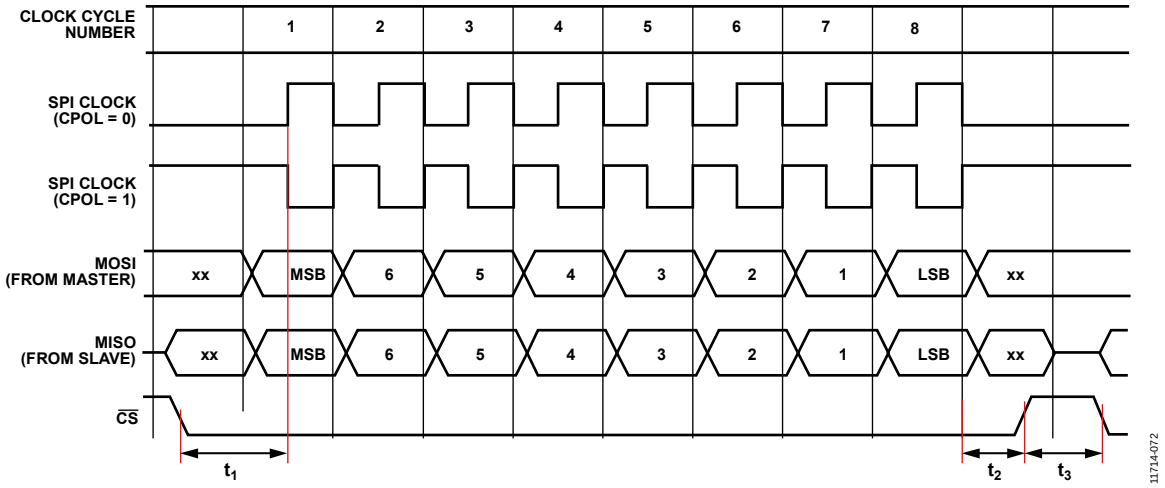


Figure 74. SPI Transfer Protocol, CPHA = 0

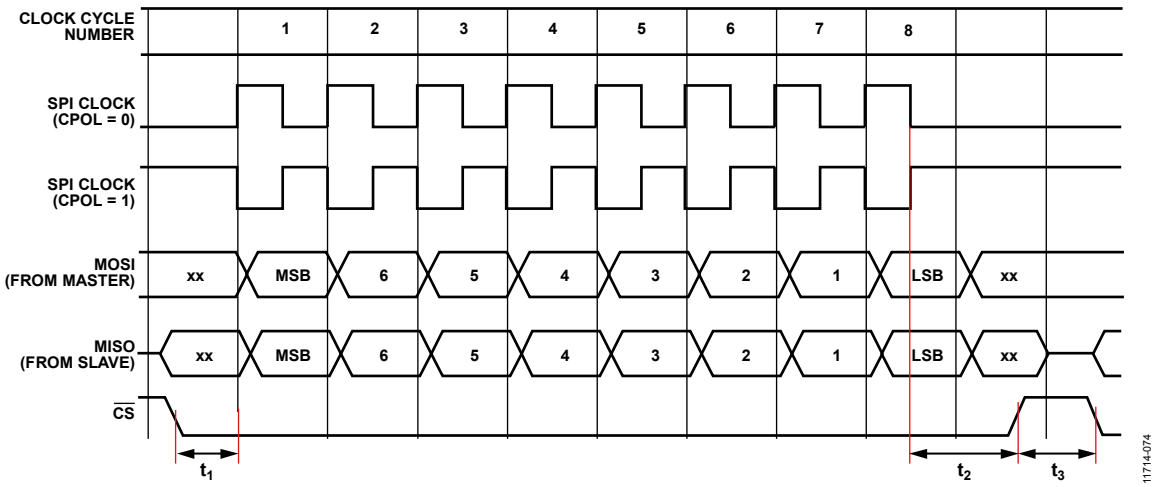


Figure 75. SPI Transfer Protocol, CPHA = 1

SPI Data Underflow and Overflow

If the Tx transmit zeros enable bit (ZEN) is cleared, the last stale byte is shifted out when a transfer is initiated with no valid data in the FIFO. If ZEN is set, 0s are transmitted when a transfer is initiated with no valid data in the FIFO. If the Rx overflow overwrite enable bit (RXOF) is set, the valid data in the Rx FIFO is overwritten by the new serial byte received when there is no space left in the FIFO. If RXOF is cleared, the new serial byte received is discarded when there is no space left in the FIFO.

When valid data is being overwritten in the Rx FIFO, the oldest byte is overwritten first followed by the next oldest byte and so on.

SPI INTERRUPTS

There is one interrupt line per SPI and four sources of interrupts. Bit 0 of the SPIxSTA register reflects the state of the interrupt line, and Bits[7:4] of the SPIxSTA register reflect the state of the four sources.

The SPI generates either a transmit interrupt request (TIRQ) or a receive interrupt request (RIRQ). Both interrupts cannot be enabled at the same time. The appropriate interrupt is enabled using the TIM bit in the SPIxCON register. If TIM = 1, TIRQ is enabled. If TIM = 0, RIRQ is enabled.

Tx Interrupt

If TIM is set, the Tx FIFO status causes the interrupt. MOD bits control when the interrupt occurs. Descriptions of the MOD bits settings are as follows:

- 00: An interrupt is generated after each byte that is transmitted. The interrupt occurs when the byte is read from the FIFO and written to the shift register.
- 01: An interrupt is generated after every two bytes that are transmitted.
- 10: An interrupt occurs after every third byte that is transmitted.
- 11: An interrupt occurs after every fourth byte that is transmitted.

The interrupts are generated depending on the number of bytes transmitted and not on the number of bytes in the FIFO. This is unlike the Rx interrupt, which depends on the number of bytes in the Rx FIFO and not the number of bytes received.

The transmit interrupt is cleared by a read to the status register. The status of this interrupt can be read by reading the Tx IRQ status bit (TX). The interrupt is disabled if the Tx FIFO flush enable bit (TFLUSH) is left high.

Note that a write to the control register, SPIxCON, resets the transmitted byte counter back to 0. For example, in a case where MOD is set to 0x3 and after three bytes have been transmitted, SPICON is written to, and then the Tx interrupt does not occur until another four bytes have been transmitted.

Rx Interrupt

If TIM is cleared, the Rx FIFO status causes the interrupt. Again, MOD controls when the interrupt occurs. The interrupt is cleared by a read of SPIxSTA. The status of this interrupt can be read by reading the Rx IRQ status bit (RX).

Interrupts are generated only when data is written to the FIFO. For example, if MOD is set to 0x0, an interrupt is generated after the first byte is received. When the status register is read, the interrupt is deactivated. If the byte is not read from the FIFO, the interrupt is not regenerated. Another interrupt is not generated until another byte is received in the FIFO.

The interrupt depends on the number of valid bytes in FIFO and not the number of bytes received. For example, when MOD is set to 0x1, an interrupt is generated after a byte is received when there are two or more bytes in the FIFO. The interrupt is not generated after every two bytes received.

The interrupt is disabled if RFLUSH is left high.

Underflow/Overflow Interrupts

Rx FIFO overflow (RXOF) and Tx FIFO underflow (TXUR) also generate SPI interrupts.

When a transfer starts with no data in the Tx FIFO, Tx FIFO underflow (TXUR) is set to indicate an underflow condition. This causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of MOD. This interrupt is disabled if TFLUSH is set.

If data is received when the Rx FIFO is already full, the RXOF bit of the status register goes high, indicating an overflow condition. This causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of MOD. This interrupt is disabled if RFLUSH is set.

All interrupts are cleared by a read of the status register or when the ENABLE bit of SPIxCON is deasserted. The Rx and Tx interrupts are also cleared if the relevant flush bits are asserted. Otherwise, the interrupts remain active even if the SPI is reconfigured.

WIRED-OR MODE (WOM)

To prevent contention when the SPI is used in a multimaster or multislave system, the data output pins, MOSI and MISO, can be configured to behave as open circuit drivers. An external pull-up resistor is required when this feature is selected. The WOM bit in the control register controls the pad enable outputs for the data lines.

SPI DMA

Two DMA channels are dedicated to SPI: transmit and receive. The two SPI DMA channels must be configured in the DMA controller.

It is possible to enable DMA request on one channel or on two channels at the same time by setting the DMA request bits for receive or transmit in the SPIxDMA register. If only the DMA transmit request (IENTXDMA) is enabled, the Rx FIFO overflows during the SPI transfer unless received data is read by user code, and an overflow interrupt is generated. To avoid generating overflow interrupts, the Rx FIFO flush bit (RFLUSH) must be set, or the SPI interrupt must be disabled in the NVIC of the core. If only the DMA receive request (IENRXDMA) is enabled, the Tx FIFO underflows. Again, to avoid underflow interrupt, the SPI interrupt must be disabled.

The SPI Tx and Rx interrupts are not generated when using DMA. MOD is not used in transmit mode and must be set to 00 in receive mode.

The DMA bit (IENTXDMA) controls the start of a DMA transfer. DMA requests are only generated when DMA enable = 1. At the end of a DMA transfer, that is, when receiving a DMA SPI transfer interrupt, this bit needs to be cleared to prevent extra DMA requests to the µDMA controller. The data still present in the Tx FIFO is transmitted if in Tx mode.

All DMA data transfers are 16-bit transfers, and the DMA must be programmed accordingly. For example, if 16 bytes of data are to be transferred over the SPI, the DMA must be programmed to perform eight 16-bit transfers. If 17 bytes are to be transferred, nine 16-bit transfers are required, and the additional byte is discarded. Data errors occur if the DMA transfers are programmed as byte wide transfers.

DMA Master Transmit Configuration

To perform SPI DMA master transmit,

1. Configure the DMA SPI Tx channel.
2. Configure the NVIC to enable DMA Tx master interrupt.
3. Configure the SPI block as follows:
    a. SPIxDIV = application dependent settings to configure serial clock frequency.
    b. SPIxCON = 0x1043; enables SPI in master mode and transmit mode, Rx FIFO flush enabled.
    c. SPIxCNT = application dependent settings to set the number of bytes to transfer. Resolves odd byte transfer requirements when used with the even byte transfer size of the DMA.
    d. SPIxDMA = 0x1; (optional) enables the FIFO to accept 16-bit core data writes.
    e. SPIxTX = 0xXXXX; (optional) up to four 16-bit core writes can be performed to preload the FIFO.
    f. SPIxDMA = 0x3; enable DMA mode, enable Tx DMA request.

All DMA transfers are expected to be 16-bit transfers. When all data present in the DMA buffer are transmitted, the DMA generates an interrupt. User code disables the DMA request. Data is still in the Tx FIFO because the DMA request is generated whenever there is free space in the Tx FIFO to keep the FIFO always full.

DMA Master Receive Configuration

The SPIxCNT register sets the number of receive bytes required by the SPI master or the number of clocks that the master must generate. When the required number of bytes has been received, no more transfers are initiated. To initiate a DMA master receive transfer, a dummy read must be performed by user code. This dummy read must not be added to the SPIxCNT number.

The counter counting the bytes as they are received is reset when SPI is disabled either by using the ENABLE bit in SPIxCON, or if SPIxCNT register is modified, by user code.

To perform SPI DMA master receive,

1. The DMA SPI Rx channel must be configured.
2. The NVIC of the core must be configured to enable DMA Rx master interrupt.
3. The SPI block must be configured as follows:
    - SPIxDIV = SPI_SERIAL_FREQ; a divided down version of PCLK that configures the serial clock frequency.
    - SPIxCON = 0x2003; enables SPI in master mode and receive mode, 1 byte transfer.
    - SPIxDMA = 0x5; enable DMA mode, enable Rx DMA request.
    - SPIxCNT = XXX; number of bytes to be received.
    - A = SPIxRX; dummy read.

The DMA transfer stops when the appropriate number of clock cycles has been generated. All DMA data transfers are 16-bit transfers, and the DMA must be programmed accordingly. For example, if 16 bytes of data are to be transferred over the SPI, the DMA must be programmed to perform eight 16-bit transfers. If 17 bytes are to be transferred, nine 16-bit transfers are required. The additional byte is padded for the final DMA transfer. Data errors occur if the DMA transfers are programmed as byte wide transfers.

Note that the DMA buffer must be of the same size as SPIxCNT (or the same size plus one if SPIxCNT is odd) to generate a DMA interrupt when the transfer is complete.

SPI AND POWER-DOWN MODES

In master mode, before entering power-down mode, it is recommended to disable the SPI block using the ENABLE bit of the SPIxCON register.

In slave mode, in either mode of operation (interrupt driven or DMA), the $\overline{CS}$ line level must be checked via the GPIO registers to ensure that the SPI is not communicating, and the SPI block must be disabled while the $\overline{CS}$ line is high.

At power-up, the SPI block can be reenabled.

SPIH vs. SPI0/SPI1

Each SPI has identical functionality, though SPIH is capable of higher transfer rates. From an SPI programming and users model perspective, SPIH, SPI0, and SPI1 are identical. The main difference between SPIH and SPI0/SPI1 is the internal bus interface that they are connected to. SPIH is connected to a higher performance APB, which is always clocked at the higher system clock rate (HCLK) and contains fewer modules requiring arbitration. SPI0 and SPI1 are connected to the main advanced peripheral bus (APB), which selectively can be clocked at a lower rate (PCLK) and whose latency is more uncertain due to a greater number of modules requiring arbitration. This means that under higher data rates, SPIH can move data more efficiently and with lower latency. SPIH is recommended for use with high data rate peripherals.

SPI MEMORY MAPPED REGISTERS

SPI Register Map

Table 408. SPI0 Peripheral Memory Register Summary

Address	Name	Description	Reset	RW
0x40004000	SPI0STA	Status	0x0000	R
0x40004004	SPI0RX	Receive	0x0000	R
0x40004008	SPI0TX	Transmit	0x0000	W
0x4000400C	SPI0DIV	Baud rate selection	0x0000	RW
0x40004010	SPI0CON	SPI configuration	0x0000	RW
0x40004014	SPI0DMA	SPI DMA enable	0x0000	RW
0x40004018	SPI0CNT	Transfer byte count	0x0000	RW

Table 409. SPI1 Peripheral Memory Register Summary

Address	Name	Description	Reset	RW
0x40004400	SPI1STA	Status	0x0000	R
0x40004404	SPI1RX	Receive	0x0000	R
0x40004408	SPI1TX	Transmit	0x0000	W
0x4000440C	SPI1DIV	Baud rate selection	0x0000	RW
0x40004410	SPI1CON	SPI configuration	0x0000	RW
0x40004414	SPI1DMA	SPI DMA enable	0x0000	RW
0x40004418	SPI1CNT	Transfer byte count	0x0000	RW

Table 410. SPIH Peripheral Memory Register Summary (Base Address 0x40024000)

Address	Name	Description	Reset	RW
0x40024000	SPIH0STA	Status	0x0000	R
0x40024004	SPIH0RX	Receive	0x0000	R
0x40024008	SPIH0TX	Transmit	0x0000	W
0x4002400C	SPIH0DIV	Baud rate selection	0x0000	RW
0x40024010	SPIH0CON	SPI configuration	0x0000	RW
0x40024014	SPIH0DMA	SPI DMA enable	0x0000	RW
0x40024018	SPIH0CNT	Transfer byte count	0x0000	RW

SPI0 (Base Address 0x40004000) registers are shown here as an example. SPI1 and SPIH are similar requiring only a change to the base address.

Status Register

Address: 0x40004000, Reset: 0x0000, Name: SPI0STA

Table 411. Bit Descriptions for SPI0STA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	CSRSG	Detected a rising edge on $\overline{CS}$, in continuous mode. This bit indicates that there was a rising edge in $\overline{CS}$ line, when the device was in continuous mode and CSIRQ_EN was asserted. This bit causes an interrupt and it is cleared when the status register is read. This can be used to identify the end of an SPI data frame.	0x0	RC
13	CSFLG	Detected a falling edge on $\overline{CS}$, in continuous mode. This bit indicates that there was a falling edge in $\overline{CS}$ line, when the device was in continuous mode and CSIRQ_EN was asserted. This bit causes an interrupt and it is cleared when the status register is read. This can be used to identify the start of an SPI data frame.	0x0	RC
12	CSERR	Detected a $\overline{CS}$ error condition. This bit indicates that the $\overline{CS}$ line was deasserted abruptly, even before the full byte of data was transmitted completely. This bit causes an interrupt and it is cleared when the status register is read.	0x0	RC

Bits	Bit Name	Description	Reset	Access
11	RXS	SPI Rx FIFO excess bytes present. This bit is set when there are more bytes in the Rx FIFO than indicated in the SPIRXMDE bits in SPICON. This bit is cleared when the number of bytes in the FIFO is equal or less than the number in SPIRXMDE.	0x0	R
[10:8]	RXFSTA	SPI Rx FIFO status. 000: Rx FIFO empty. 001: 1 valid byte in FIFO. 010: 2 valid bytes in the FIFO. 011: 3 valid bytes in the FIFO. 100: 4 valid bytes in the FIFO.	0x0	R
7	RXOF	SPI Rx FIFO overflow. Set when the Rx FIFO was already full when new data was loaded to the FIFO. This bit generates an interrupt except when RFLUSH is set in SPICON. Cleared when the SPISTA register is read.	0x0	RC
6	RX	SPI Rx IRQ. Not available in DMA mode. Set when a receive interrupt occurs. This bit is set when TIM in SPICON is cleared and the required number of bytes has been received. Cleared when the SPISTA register is read.	0x0	RC
5	TX	SPI Tx IRQ status bit. Not available in DMA mode. Set when a transmit interrupt occurs. This bit is set when TIM in SPICON is set and the required number of bytes has been transmitted. Cleared when the SPISTA register is read.	0x0	RC
4	TXUR	SPI Tx FIFO underflow. This bit is set when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt except when TFLUSH is set in SPICON. Cleared when the SPISTA register is read.	0x0	RC
[3:1]	TXFSTA	SPI Tx FIFO status. 000: Tx FIFO empty. 001: 1 valid byte in FIFO. 010: 2 valid bytes in FIFO. 011: 3 valid bytes in FIFO. 100: 4 valid bytes in FIFO.	0x0	R
0	IRQ	SPI interrupt status. Set to 1 when an SPI based interrupt occurs. Cleared after reading SPISTA.	0x0	RC

Receive Register

Address: 0x40004004, Reset: 0x0000, Name: SPI0RX

Table 412. Bit Descriptions for SPI0RX

Bits	Bit Name	Description	Reset	Access
[15:8]	DMA_DATA_BYTE_2	8-bit receive buffer. These eight bits are used only in DMA mode, where all FIFO accesses happen as half word access. They return zeros if DMA is disabled.	0x0	R
[7:0]	DATA_BYTE_1	8-bit receive buffer.	0x0	R

Transmit Register

Address: 0x40004008, Reset: 0x0000, Name: SPI0TX

Table 413. Bit Descriptions for SPI0TX

Bits	Bit Name	Description	Reset	Access
[15:8]	DMA_DATA_BYTE_2	8-bit transmit buffer. These eight bits are used only in the DMA mode, where all FIFO accesses happen as half word access. They return zeros if DMA is disabled.	0x0	W
[7:0]	DATA_BYTE_1	8-bit transmit buffer.	0x0	W

Baud Rate Selection Register

Address: 0x4000400C, Reset: 0x0000, Name: SPI0DIV

Table 414. Bit Descriptions for SPI0DIV

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	CSIRQ_EN	Enable interrupt on every $\overline{CS}$ edge in continuous mode. If this bit is set and the SPI module is in continuous mode, any edge on $\overline{CS}$ generates an interrupt and the corresponding status bits (CSRSG, CSFLG) are asserted. If this bit is clear, no interrupt is generated. This bit has no effect if the SPI is not in continuous mode.	0x0	RW
7	MD_CS_RST	Reset mode for CSERR. If this bit is set, the bit counter is reset after a $\overline{CS}$ error condition and the Cortex-M3 is expected to clear the SPI ENABLE bit. If this bit is clear, the bit counter continues from where it stopped. SPI can receive the remaining bits when $\overline{CS}$ gets asserted and Cortex-M3 has to ignore the CSERR interrupt. However, it is strongly recommended to set this bit for a graceful recovery after a $\overline{CS}$ error.	0x0	RW
6	HFM	High frequency mode. This bit is used for applications using high frequency where the pad introduces a significant delay on the SCL. This can cause a significant enough difference between the serial clock and the data being received on the Rx shift register. In this mode, the Rx shift register is clocked by SCLIN instead of UCLK.	0x0	RW
[5:0]	DIV	SPI clock divider. DIV is the factor used to divide UCLK to generate the serial clock.	0x0	RW

SPI Configuration Register

Address: 0x40004010, Reset: 0x0000, Name: SPI0CON

Table 415. Bit Descriptions for SPI0CON

Bits	Bit Name	Description	Reset	Access
[15:14]	MOD	SPI IRQ mode bits. These bits configure when the Tx/Rx interrupts occur in a transfer. For DMA Rx transfer, these bits must be 00. 00: Tx interrupt occurs when one byte has been transferred. Rx interrupt occurs when one or more bytes have been received into the FIFO. 01: Tx interrupt occurs when two bytes have been transferred. Rx interrupt occurs when two or more bytes have been received into the FIFO. 10: Tx interrupt occurs when three bytes have been transferred. Rx interrupt occurs when three or more bytes have been received into the FIFO. 11: Tx interrupt occurs when four bytes have been transferred. Rx interrupt occurs when the Rx FIFO is full, or four bytes are present.	0x0	RW
13	TFLUSH	SPI Tx FIFO flush enable. Set this bit to flush the Tx FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or 0x00 is transmitted depending on the ZEN bit. Any writes to the Tx FIFO are ignored while this bit is set. Clear this bit to disable Tx FIFO flushing.	0x0	RW
12	RFLUSH	SPI Rx FIFO flush enable. Set this bit to flush the Rx FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is set, all incoming data is ignored and no interrupts are generated. If this bit is set and TIM = 0, a read of the Rx FIFO initiates a transfer. Clear this bit to disable Rx FIFO flushing.	0x0	RW
11	CON	Continuous transfer enable. Set by user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. $\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty. Cleared by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register, a new transfer is initiated after a stall period of one serial clock cycle.	0x0	RW
10	LOOPBACK	Loopback enable. Set by user to connect MISO to MOSI and test software. Cleared by user to be in normal mode.	0x0	RW
9	OEN	Slave MISO output enable. Set this bit for MISO to operate as normal. Clear this bit to disable the output driver on the MISO pin. The MISO pin is open circuit when this bit is clear.	0x0	RW
8	RXOF	SPIRX overflow overwrite enable. Set by user, the valid data in the Rx register is overwritten by the new serial byte received. Cleared by user, the new serial byte received is discarded.	0x0	RW
7	ZEN	Transmit zeros enable. Set this bit to transmit 0x00 when there is no valid data in the Tx FIFO. Clear this bit to transmit the last transmitted value when there is no valid data in the Tx FIFO.	0x0	RW

Bits	Bit Name	Description	Reset	Access
6	TIM	SPI transfer and interrupt mode. Set by user to initiate transfer with a write to the SPITX register. Interrupt only occurs when Tx is empty. Cleared by user to initiate transfer with a read of the SPIRX register. Interrupt only occurs when Rx is full.	0x0	RW
5	LSB	LSB first transfer enable. 0: MSB transmitted first. 1: LSB transmitted first.	0x0	RW
4	WOM	SPI wired OR mode. 1: enables open circuit data output enable. External pull-ups required on data out pins. 0: normal output levels.	0x0	RW
3	CPOL	Serial clock polarity. 0: serial clock idles low. 1: serial clock idles high.	0x0	RW
2	CPHA	Serial clock phase mode. 1: serial clock pulses at the beginning of each serial bit transfer. 0: serial clock pulses at the end of each serial bit transfer.	0x0	RW
1	MASEN	Master mode enable. 0: enable slave mode. 1: enable master mode.	0x0	RW
0	ENABLE	SPI enable. 0: disable the SPI. 1: enable the SPI.	0x0	RW

SPI DMA Enable Register

Address: 0x40004014, Reset: 0x0000, Name: SPI0DMA

Table 416. Bit Descriptions for SPI0DMA

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
2	IENRXDMA	Enable receive DMA request. 0: receive DMA requests are disabled. 1: receive DMA requests are enabled.	0x0	RW
1	IENTXDMA	Enable transmit DMA request. 0: transmit DMA requests are disabled. 1: transmit DMA requests are enabled.	0x0	RW
0	ENABLE	Enable DMA for data transfer. Set by user code to start a DMA transfer. Cleared by user code at the end of DMA transfer. This bit must be cleared to prevent extra DMA request to the µDMA controller. 0: DMA transfers are disabled. 1: DMA transfers are enabled.	0x0	RW

Transfer Byte Count Register

Address: 0x40004018, Reset: 0x0000, Name: SPI0CNT

Table 417. Bit Descriptions for SPI0CNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	COUNT	Transfer byte count. COUNT indicates the number of bytes to be transferred. COUNT is used in both receive and transmit transfer types. The COUNT value assures that a master mode transfer terminates at the proper time and that 16-bit DMA transfers are byte padded or discarded as required to match odd transfer counts. Reset by clearing Bit 0 of SPICON or if SPI0CNT is updated.	0x0	RW

I²S INTERFACE

FEATURES

Inter-IC Sound (I²S) bus is an industry standard synchronous serial interface established specifically for digital audio communication. The standard implements a 3-wire stereo transmission of digital audio. It supports word lengths of up to 24 bits and audio sampling frequencies of up to 192 kHz.

The purpose of the I²S port on the ADuCM350 is to provide audio data to an amplifier, which drives a small speaker. In this capacity, the included I²S port is a transmit only port.

The I²S port is connected to the system via the advanced peripheral bus (APB). It also incorporates a DMA channel that interfaces with the µDMA controller of the Cortex-M3.

The I²S features available on ADuCM350 include the following:

- Philips compliant stereo serial transmission.
- Data samples of up to 24 bits.
- Frame clocks from 8 kHz to 192 kHz.
- Master/slave mode.
- Tx FIFO depth of 8 samples.
- DMA mode, with address auto increment.
- Interrupt mode.
- Downsampling transfers.

MODULE ARCHITECTURE

Figure 76 shows a top level diagram of the I²S port. The module consists of an APB interface with a Tx FIFO depth of eight samples and the I²S serial to parallel data converter. There is also an interface to the DMA and Cortex-M3 core interrupts.

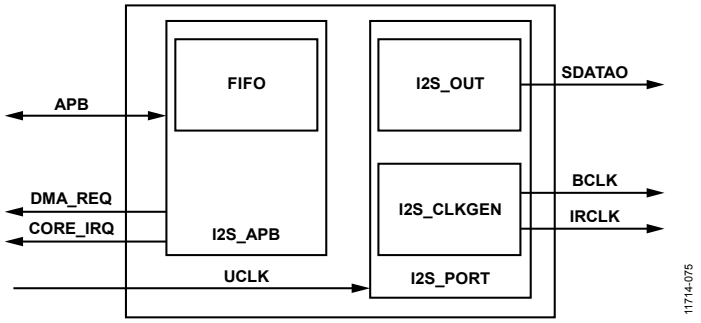


Figure 76. Module Block Diagram

The module operates in three clock domains as follows:

- PCLK clocks the APB interface.
- UCLK clocks the I²S state machines and is synchronous to PCLK. It can, however, be gated independently.
- BCLK is the I²S bit clock and clocks the data out.

OPERATION

The port can be configured for master or slave operation. The master is defined as the device that is generating the bit clock and the frame clock. In master mode, the module generates its own frame clock (LRCLK) and bit clock (BCLK) from a master clock. In slave mode, the frame clock and bit clock are supplied externally. The master/slave mode is set by the SAI_MS bit in the I2S_MODE1 register. The module is a transmitter and cannot receive I²S data.

I²S INTERFACE

The module interface consists of nine pins, and the chip level interface consists of three pads. Table 418 summarizes the interface.

Table 418. I²S Interface Ports

Description	Pin	CSPBGA BUMP LOCATION	Direction
Serial Data Output	SDATA	K7	Output
Bit Clock Output	BMCLK	K6	Output
Frame Clock Output	LRCLK	J6	Output

Data Out (SDATA)

SDATA is the serial data pin. Serial data is transmitted in twos complement with the MSB being sent first (as per the Philips standard). The SAI_MSB bit in the I2S_MODE1 register can be set to transmit with the LSB first. The data can be delayed by one clock (as per the Philips standard) but can also be left or right justified to the frame by programming the SDATA_FMT bits in the I2S_MODE1 register. The data width can be programmed in 16- or 24-bit mode (using the DATA_WIDTH bit in the I2S_MODE1 register), although the data itself can be 16 bits or less in 16-bit mode and between 17 bits and 24 bits in 24-bit mode. The internal datapath is 24 bits wide. Unused bits can be driven low or left floating by setting the DRV_HIZ bit in the I2S_MODE1 register. The DATA_WIDTH bit is useful for trimming unused data remnants in the transmit FIFO when the data width is switched from 24-bit serial data to 8- or 16-bit serial data.

Frame Clock (LRCLK)

LRCLK is the frame clock pin. The frame clock indicates the channel being transmitted:

- LRCLK = 0; Channel 1 (left, mono).
- LRCLK = 1; Channel 2 (right).

The frame edges are used to synchronize the data between the transmitter and the receiver. A frame edge indicates the start of a new data word, regardless of word length. The frame clock can change on either the rising or falling edge of the bit clock. In the slave, the signal is latched on the rising edge of the bit clock. The frame clock polarity can be programmed using the LR_POL bit in the I2S_MODE2 register.

Bit Clock (BMCLK)

BMCLK is the bit clock pin. Data changes on the rising or falling edges of the bit clock. The bit clock edge can be set with the BCLK_EDGE bit in the I2S_MODE2 register. By default, data is transmitted on the falling edge of the bit clock. In master mode, the bit clock can extend for 16 or 32 cycles for each frame. This is specified using the BCLK_RATE bit in the I2S_MODE1 register when the module operates in master mode. The I²S standard itself does not specify a particular frame length, and frames (and data words) can be expected to be of any length.

Figure 77 shows the timing diagram of a typical I²S stereo transfer from the Philips I²S specification. WS (word select) is the frame clock and SCK (serial clock) is the bit clock.

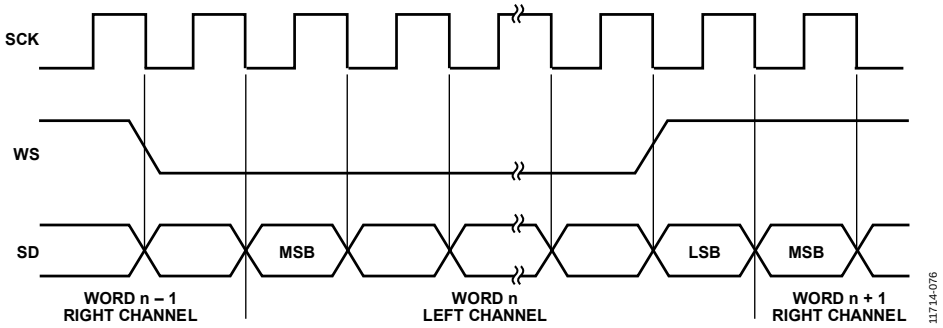


Figure 77. I²S Timing Diagram

I²S MODES

The I²S module supports various formatting modes, which are described in Table 419 and Figure 78.

Table 419. I²S Modes

	True I²S	Register
Frame Width	Slave: any; master: 16 or 32	I2S_MODE1: BCLK_RATE
Data Width	Slave: up to 24; master: 16 or 24	I2S_MODE1: DATA_WIDTH
Format	1 cycle delay, left justified (LJ24), right justified (RJ24)	I2S_MODE1: SDATA_FMT
Frame Edge	BMCLK fall	Not applicable
Data Edge	BMCLK fall/rise	I2S_MODE2: BCLK_EDGE

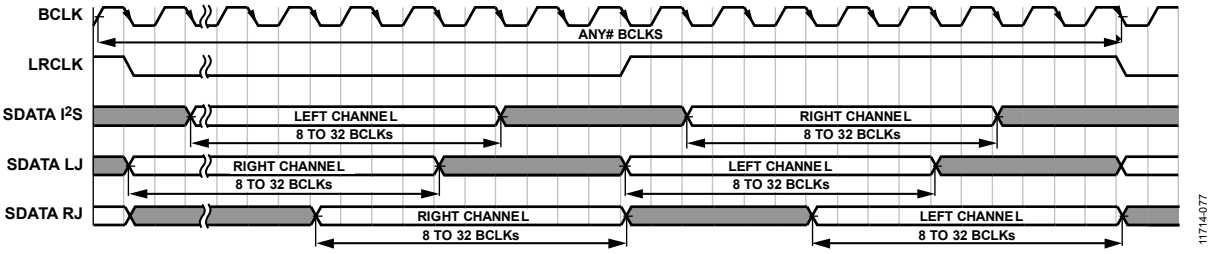


Figure 78. I²S Format Modes

CLOCK GENERATOR

When the port operates in slave mode, the bit clock and frame clock are received externally. When the port operates in master mode, the bit clock and frame clock are generated internally from the 16 MHz internal oscillator or external crystal. The clock frequencies generated are based on the I²S port settings SAI (number of channels on the I²S bus), BCLK_RATE (number of BMCLKs per channel), and FS (frame rate).

I²S TRANSFER INITIATION

For any transmission to occur, the I2S_EN bit in the I2S_CFG1 register must be set. Transmit samples are stored in the I2S_OUT registers. The 16 MSBs of each sample are stored in the OUTxH registers, and the 8 LSBs are stored in OUTxL registers. In the case of the 16-bit samples, the entire sample is stored in the OUTxH registers. Notice that because the APB is 16 bits wide, it requires 2 APB transfers for a 24-bit sample. A 16-bit sample only requires 1 APB transfer. Single 16-bit sample transfers are enabled by setting the TRAN_MODE bits in the I2S_CFG1 register. Writes to unused data registers are not supported; therefore, if streaming only 8-bit or 16-bit data, the user code must not write to the OUTxL registers.

A transfer is started when a sample is written to the I2S_OUT registers. The CHAN_SEL1 and CHAN_SEL2 bits in the I2S_CFG1 register indicate which I2S_OUT registers must be populated for a transfer to start. Refer to Table 420 for details.

Table 420. Transfer Initiation

		Description of the I2S_OUTx Registers	
CHAN_SEL1	**CHAN_SEL2**	**16-Bit Mode**	**24-Bit Mode**
0	0	No transfer can start even if the I2S_OUT registers are written via the APB.	No transfer can start even if the I2S_OUT registers are written via the APB.
0	1	A transfer starts when OUT2H is written.	A transfer starts when both OUT2H and OUT2L are written.
1	0	A transfer starts when OUT1H is written.	A transfer starts when both OUT1H and OUT1L are written.
1	1	A transfer starts when both OUT2H and OUT1H are written.	A transfer starts when all OUT1H, OUT1L, OUT2H, and OUT2L are written.

I²S INTERRUPTS

The module can interrupt the core via its single interrupt line (CORE_IRQ, active high). Interrupts are enabled via the REQ_EN and STAT_EN bits in the I2S_CFG1 register. An interrupt is generated from either of two sources: a transmit data request or an error due to a FIFO underflow or overflow.

Tx Interrupt

The transmit FIFO generates interrupts indicated by the REQ_PEND bit in the I2S_STAT register. A 1 indicates an interrupt is pending. Writing a 1 to this bit clears the interrupt.

By default, the FIFO interrupts the core after every transmitted sample. To reduce the demands on the core, the REQ_FREQ bits in the I2S_CFG2 register can be programmed as follows:

- 00: send an interrupt to the core for every sample that is transmitted.
- 01: send an interrupt to the core for every four samples that are transmitted.
- 10: send an interrupt to the core for every eight samples that are transmitted.
- 11: send an interrupt to the core for every 16 samples that are transmitted.

It is assumed that if the core is interrupted every n samples (starting with the first sample after the I²S port is enabled), n samples are written to transmit the FIFO every interrupt. Because the FIFO is 8 samples deep, setting the interrupt to fire every 16 samples requires the SAMP_FREQ bits in the I2S_CFG2 register to be set to downsample the incoming data according to the following description:

- 00: write to serial output every sample.
- 01: write to serial output every four samples.
- 10: write to serial output every eight samples.
- 11: write to serial output every 16 samples.

Downsampling can be set independent of interrupts if the application requires it. It must be set if the interrupts are asserted at a rate lower than the FIFO depth.

Error Interrupt

An error can occur when the transmit FIFO is full or empty. Specifically, when a sample is written to a full FIFO or is transmitted from an empty FIFO, an error is flagged. The STAT_PEND bit in the I2S_STAT register is set to 1 when an error is flagged. When interrupts are enabled, the error generates an interrupt to the core. The error can be cleared by writing a 1 to the same location.

The FIFO_STAT bits in the I2S_STAT register specify the number of samples that are currently in the FIFO. This can be useful in determining the number of samples that can be written to the FIFO. FIFO_STAT can also be useful in determining why the STAT_PEND bit is set. STAT_PEND = 1 while FIFO_STAT = 0x0 indicates an underflow condition, whereas STAT_PEND = 1 while FIFO_STAT = 0x8 indicates a write (by core or DMA) has been made to a FIFO that was full. The core is expected to take the appropriate action to handle the error and clear the error bits to resume normal operation.

Note that when the user wishes to stop I²S transfers, the I2S_EN bit in the I2S_CFG1 register must be deasserted after the last sample has been written in the I2S_OUT registers. Otherwise, the FIFO flags an empty error after the last sample has been transmitted.

I²S DMA

One DMA channel is dedicated to the I²S data transmission. The I²S DMA must be configured in the DMA controller.

To enable DMA requests from the I²S module, the DMA_EN bit in the I2S_CFG1 register is set. Similar to requests from the APB, DMA requests can be done every sample or less frequently by programming the DMA_FREQ bits in the I2S_CFG2 register.

- 00: send a DMA request when FIFO has one or more spaces available.
- 01: send a DMA request when FIFO has four or more spaces available.
- 10: send a DMA request when FIFO has eight or more spaces available.

Address Auto-Increment

The DMA engine works best when the transfer destination is a single address register. However, a full I²S stereo transmission requires accessing up to four separate addresses (the four I2S_OUT registers). For the DMA to service all four locations, the memory pointer must be reset every frame, reducing the efficiency of the DMA transfer.

To avoid this situation, the AUTO_INCR bit in the I2S_CFG1 register can be set. When this feature is enabled for Port 1 or both ports, the DMA can write a single address (I2S_OUT1L for 24-bit data, I2S_OUT1H for 16-bit data), and the module then relocates the data-words to the appropriate addresses internally. If only Port 2 is being used, the DMA can write a single address (I2S_OUT2L for 24-bit data, I2S_OUT2H for 16-bit data), and the module then relocates the data-words to the appropriate addresses internally.

Therefore, a block of interleaved audio samples in memory can fan out correctly to the I²S module via the DMA without the DMA engine knowing that the data must be in four different addresses.

Note that data in memory must be left and right channel interleaved, with the first sample being for the left channel.

R_POWER Selection

Note that the I²S peripheral is the only peripheral where the DMA R_POWER setting can be a value other than b0000. Table 421 lists the R_POWER setting for the different I²S formats. See the DMA Controller section for more information.

Table 421. R_POWER Selection

Transfer Mode	DMA R_POWER Setting
16-Bit Mono	b0000
16-Bit Stereo	b0001
16-Bit Mono	b0001
16-Bit Stereo	b0010

I²S DRIVER EXAMPLE CONFIGURATION

The I²S driver operates the SSM2519 Class-D audio power amplifier.

For the example configuration, the following settings were selected:

- SSM2519 set up in I²C mode, not in standalone mode.
- ADuCM350 set up with standard I²S delay by 1 data format (using the I2S_MODE1 register, Bits[15:14]), which is the default alignment setting on both ICs.
- The SSM2519 is used in the no BCLK operation mode (see the SSM2519 data sheet for more information about the register setting). The ADuCM350 I²S bit clock signal is used to drive the SSM2519 MCLK; therefore, no separate clock source is required. The I²S driver is specifically designed to generate the clock rates required for this mode of operation.
- To generate the high I²S bit clock rate requirements, the I²S must be configured as TDM16. The frame rate must be configured to 8 kHz in the I2S_MODE1 register, Bits[10:8], and to 16-bit slot width in the I2S_MODE1 register, Bits[7:6]. This equates to an I²S bit clock of 8 kHz × 16 × 16 = 2.048 MHz, which is the minimum MCLK for the SSM2519.
- The unused bit streams must be put in a high impedance state in the I2S_MODE1 register, Bit 0.

I²S MEMORY MAPPED REGISTERS

I²S Register Map

Table 422. I²S Register Summary

Address	Name	Description	Reset	RW
0x40005800	I2S_OUT1L	Channel 1 LSBs	0x0000	RW
0x40005804	I2S_OUT1H	Channel 1 MSBs	0x0000	RW
0x40005808	I2S_OUT2L	Channel 2 LSBs	0x0000	RW
0x4000580C	I2S_OUT2H	Channel 2 MSBs	0x0000	RW
0x40005810	I2S_MODE1	I²S Format Modes 1	0x0201	RW
0x40005814	I2S_MODE2	I²S Format Modes 2	0x3100	RW
0x40005818	I2S_CFG1	I²S Configuration 1	0x0000	RW
0x4000581C	I2S_CFG2	I²S Configuration 2	0x0000	RW
0x40005820	I2S_STAT	I²S status	0x0000	RW

Channel 1 LSBs Register

Address: 0x40005800, Reset: 0x0000, Name: I2S_OUT1L

Table 423. Bit Descriptions for I2S_OUT1L

Bits	Bit Name	Description	Reset	Access
[15:8]	OUT1_LSBS	LSBs of Channel 1 output. Serial Output Data Channel 1, Bits[7:0]. Writes to this register are not supported in 8- or 16-bit transaction modes, or when Channel 1 is not enabled.	0x0	W
[7:0]	RESERVED	Reserved. Always returns 0 when read.	0x0	R

Channel 1 MSBs Register

Address: 0x40005804, Reset: 0x0000, Name: I2S_OUT1H

Table 424. Bit Descriptions for I2S_OUT1H

Bits	Bit Name	Description	Reset	Access
[15:0]	OUT1_MSBS	MSBs of Channel 1 output. Serial Output Data Channel 1, Bits[23:8]. If 16-bit samples are used, the entire sample, Bits[15:0], is stored in this register. Writes to this register are not supported when Channel 1 is not enabled.	0x0	W

Channel 2 LSBs Register

Address: 0x40005808, Reset: 0x0000, Name: I2S_OUT2L

Table 425. Bit Descriptions for I2S_OUT2L

Bits	Bit Name	Description	Reset	Access
[15:8]	OUT2_LSBS	LSBs of Channel 2 output. Serial Output Data Channel 2, Bits[7:0]. Writes to this register are not supported in 8- or 16-bit transaction modes, or when Channel 2 is not enabled.	0x0	W
[7:0]	RESERVED	Reserved. Always returns 0 when read.	0x0	R

Channel 2 MSBs Register

Address: 0x4000580C, Reset: 0x0000, Name: I2S_OUT2H

Table 426. Bit Descriptions for I2S_OUT2H

Bits	Bit Name	Description	Reset	Access
[15:0]	OUT2_MSBS	MSBs of Channel 2 output. Serial Output Data Channel 2, Bits[23:8]. If 16-bit samples are used, the entire sample, Bits[15:0], is stored in this register. Writes to this register are not supported when Channel 2 is not enabled.	0x0	W

I²S Format Modes 1 Register

Address: 0x40005810, Reset: 0x0201, Name: I2S_MODE1

The MODE registers describe the operating mode of the I²S SDATA stream.

Table 427. Bit Descriptions for I2S_MODE1

Bits	Bit Name	Description	Reset	Access
[15:14]	SDATA_FMT	Data format. 00: I²S default SDATA format (delay by one BMCLK cycle). 01: left justified SDATA format. 10: 24-bit right justified SDATA format. Note that 32 BMCLKs per channel are required in stereo mode. 11: 16-bit right justified SDATA format. Note that 32 BMCLKs per channel required in stereo mode.	0x0	RW
[13:11]	SAI	Channel format. 000: stereo mode operation. 001: TDM2 mode operation. 010: TDM4 mode operation. 011: TDM8 mode operation. 100: TDM16 mode operation. 101: mono/PCM mode operation. 110 to 111: reserved.	0x0	RW
[10:8]	FS	Frame rate. Defines the frame rate. Nominal values assume a PCLK at 16 MHz. 000: 8 kHz nominal frame rate (PCLK/2048). 001: 16 kHz nominal frame rate (PCLK/1024). 010: 32 kHz nominal frame rate (PCLK/512). 011: 64 kHz nominal frame rate (PCLK/256). 100: 128 kHz nominal frame rate (PCLK/128). 101 to 111: reserved.	0x2	RW
[7:6]	SLOT_WIDTH	Sloth width. 00: 32-bit channel slots in TDM mode. 01: 24-bit channel slots in TDM mode. 10: 16-bit channel slots in TDM mode. 11: reserved.	0x0	RW
5	DATA_WIDTH	Data width. Setting this bit trims the data stream to fit in 16 bits per channel. This bit has no effect on data streams that are already 16 bits or fewer. 0: 24-bit data output. 1: 16-bit data output.	0x0	RW
4	LR_MODE	TDM clock mode. 0: LRCLK 50% duty cycle in TDM mode. 1: LRCLK single cycle pulse in TDM mode.	0x0	RW
3	SAI_MSB	SDATA endian format. 0: MSB first, LSB last format SDATA. 1: LSB first, MSB last format SDATA.	0x0	RW
2	BCLK_RATE	Bit clock rate. 0: 32 BMCLKs/channel in master mode. 1: 16 BMCLKs/channel in master mode.	0x0	RW
1	SAI_MS	Master slave select. 0: slave mode port operation (use external BMCLK, LRCLK). 1: master mode port operation (use internally generated clocks).	0x0	RW
0	DRV_HIZ	Unused bit drive option. 0: drive unused bits in SDATA stream. 1: don't drive unused bits in SDATA stream.	0x1	RW

I²S Format Modes 2 Register

Address: 0x40005814, Reset: 0x3100, Name: I2S_MODE2

The MODE registers describe the operating mode of the I²S SDATA stream.

Table 428. Bit Descriptions for I2S_MODE2

Bits	Bit Name	Description	Reset	Access
15	LR_POL	LRCLK polarity. 0: don't invert LRCLK to port. 1: invert LRCLK to port.	0x0	RW
14	BCLK_EDGE	Bit clock edge. 0: SDATA streams change on falling edge of BCLK (both input and output). 1: SDATA streams change on rising edge of BCLK (both input and output).	0x0	RW
13	DRV_CH2	Channel 2 slot drive enable. 0: don't drive Channel 2 slot on output SDATA. 1: drive Channel 2 slot on output SDATA.	0x1	RW
12	DRV_CH1	Channel 1 slot drive enable. 0: don't drive Channel 1 slot on output SDATA. 1: drive Channel 1 slot on output SDATA.	0x1	RW
[11:8]	CMAP_C2	Channel 2 output slot. Channel 2 SDATA output slot selection.	0x1	RW
[7:4]	CMAP_C1	Channel 1 output slot. Channel 1 SDATA output slot selection.	0x0	RW
[3:0]	RESERVED	Reserved. Always returns 0 when read.	0x0	R

I²S Configuration 1 Register

Address: 0x40005818, Reset: 0x0000, Name: I2S_CFG1

This register enables and disables specific features of the I²S port over the APB bus including DMA interrupts, core interrupts, and error reporting.

Table 429. Bit Descriptions for I2S_CFG1

Bits	Bit Name	Description	Reset	Access
15	FIFO_RST	FIFO reset and hold. 0: normal operation. 1: reset and hold FIFO pointers for serial output.	0x0	RW
14	INCR_RST	Address automatic increment reset. 0: normal operation. 1: reset automatic increment address.	0x0	RW
13	DMA_EN	DMA enable. 0: disable DMA requests from serial output port. 1: enable DMA requests from serial output port.	0x0	RW
12	REQ_EN	Interrupt enable for serial data requests. 0: disable data request interrupts to core from serial port. 1: enable data request interrupts to core from serial port. When enabled, an interrupt fires each time a frame starts on the I²S bus. See I2S_STAT register for interrupt flag.	0x0	RW
11	STAT_EN	Interrupt enable for FIFO status. 0: disable FIFO status reporting via core interrupt. 1: enable FIFO status reporting via core interrupt. When enabled, an interrupt fires if the serial output FIFO becomes full or empty. See I2S_STAT register for interrupt flag.	0x0	RW
10	AUTO_INCR	Addressing automatic increment. 0: standard addressing. 1: enable automatic increment addressing. A single data address can be accessed and the internal I²S data address automatically increments circularly (depending on the number of channels selected in CHAN_SELx with each subsequent access of that same base register.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[9:8]	TRAN_MODE	Transfer mode: 24-bit, 16-bit, or 8-bit. 00: 24-bit audio transfers over APB (two writes per channel). 01: 16-bit audio transfers over APB (only one write per channel). 10: 8-bit audio transfers over APB (one write per channel). In this case, the 16-bit I2S_OUTMSBS register is written as two packed 8-bit samples. The eight LSBs of the register contain sample n, and the eight MSBs of the register contain sample n + 1. In this mode, because two samples are sent per register write, the interrupts/DMA requests only fire every two I²S frames. 11: reserved.	0x0	RW
7	CHAN_SEL2	Channel 2 APB transfer. 0: Channel 2 is not transferred over APB. 1: Channel 2 is transferred over APB.	0x0	RW
6	CHAN_SEL1	Channel 1 APB transfer. 0: Channel 1 is not transferred over APB. 1: Channel 1 is transferred over APB.	0x0	RW
[5:1]	RESERVED	Always return 0 when read.	0x0	RW
0	I2S_EN	Enable I²S port. 0: I²S port is disabled. 1: I²S port is enabled.	0x0	RW

I²S Configuration 2 Register

Address: 0x4000581C, Reset: 0x0000, Name: I2S_CFG2

This register configures the effective sampling rate of the I²S port features over the APB. For example, the DMA and/or core interrupts can be configured to fire only every 4, 8, or 16 samples. This is particularly useful for doing burst style transfers. The FIFO can also be slowed down via SAMP_FREQ providing a cheap downsampling operation.

Table 430. Bit Descriptions for I2S_CFG2

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[5:4]	REQ_FREQ	Data request interrupt sampling frequency. 00: send interrupt for serial data request every frame or every other frame if 8-bit transfer mode is enabled. 01: send interrupt for serial data request every 4 frames. 10: send interrupt for serial data request every 8 frames. 11: send interrupt for serial data request every 16 frames.	0x0	RW
[3:2]	DMA_FREQ	DMA request sampling frequency. 00: send DMA request when there is at least one available space in FIFO (two spaces if in 8-bit transfer mode). 01: send DMA request when there are at least 4 available spaces in FIFO. 10: send DMA request when there are at least 8 available spaces in FIFO. 11: reserved.	0x0	RW
[1:0]	SAMP_FREQ	Serial output data sampling frequency. 00: write to serial output every frames. 01: write to serial output every 4 frames. 10: write to serial output every 8 frames. 11: write to serial output every 16 frames.	0x0	RW

I²S Status Register

Address: 0x40005820, Reset: 0x0000, Name: I2S_STAT

This register holds interrupt flags and other status bits. Interrupt flags must be cleared by writing to this register for the core IRQ signal to clear.

Table 431. Bit Descriptions for I2S_STAT

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[5:2]	FIFO_STAT	FIFO status. A 4-bit field indicating the number of samples currently in the FIFO. Initially, the FIFO is empty as indicated by FIFO_STAT = 0x0. As samples are written to the FIFO, FIFO_STAT increments with each sample. The maximum value of FIFO_STAT is 0x8, indicating eight samples are present in the FIFO. In the event a sample is written to a full FIFO, the sample is ignored and FIFO_STAT remains 0x8. 0x0: FIFO is empty. 0x1: FIFO has 1 sample. 0xn: FIFO has n samples. 0x8: FIFO is full.	0x0	R
1	REQ_PEND	Data request pending. Write 1 to this bit to clear the data request interrupt. Indicates samples are to be written into the FIFO. The rate at which this request is generated is determined by the REQ_FREQ bits located in the I2S_CFG2. 0: no data request pending for serial output port. 1: data request is pending for serial output.	0x0	RW1C
0	STAT_PEND	FIFO status error. Bit must be written with a 1 to be cleared. This bit indicates a FIFO underrun has occurred (read access of an empty FIFO) or an overrun has occurred (write to a full FIFO). 0: a FIFO access error has not been detected. 1: a FIFO underrun or overrun error has been detected.	0x0	RW1C

BEEPER DRIVER

FEATURES

The beeper driver module generates a differential square wave of programmable frequency. It is used to drive an external piezoelectric sound component whose two terminals connect to the differential square wave output. A standard APB interface connects this module to the system bus.

The beeper driver present in the ADuCM350 includes the following features:

- A module that can deliver frequencies from 8 kHz to ~0.25 kHz.
- It operates on a fixed independent 32 kHz (32,768 Hz) clock source that is unaffected by changes in system clocks.
- A timer that allows for programmable tone durations from 4 ms to 1.02 sec in 4 ms increments.
- Single tone (pulse) and multitone (sequence) modes provide versatile playback options.
- In sequence mode, the beeper can be programmed to play any number of tone pairs from 1 to 254 (2 tones to 508 tones) or can be programmed to play indefinitely (until stopped by the user).
- Any tone can be programmed with an infinite duration, to play continuously until stopped by the user.
- Interrupts are available to indicate the start or end of any beep, the end of a sequence, or that the sequence is nearing completion.

MODULE ARCHITECTURE

A block diagram of the beeper driver is shown in Figure 79.

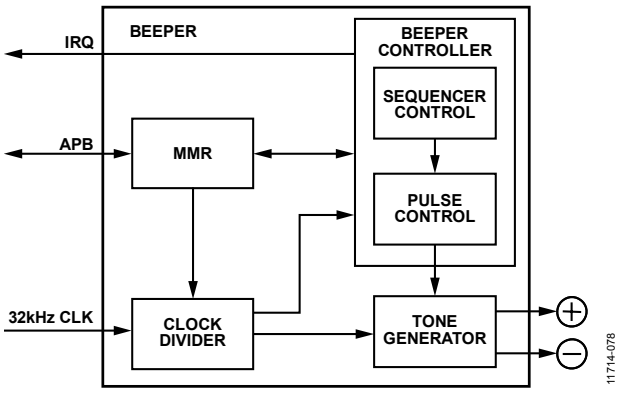


Figure 79. Beeper Driver Block Diagram

The registers of the module (MMR block) are programmed via the APB protocol of the AMBA. The clock divider divides the 32 kHz input clock down to frequencies between 8 kHz and ~0.25 kHz for driving the off-chip piezoelectric component. The beeper controller block controls the tone generator, enabling and disabling its operation, and selecting the appropriate frequency output from the clock divider. The beeper controller also tracks the durations and repeating sequences of tones. The beeper module can interrupt the microcontroller with one interrupt line. Its off chip output lines are held low when the module is disabled (no voltage across the piezoelectric component) and are used to drive a differential square wave when tones are being played.

OPERATION

The basic operation of the beeper driver consists of writing tone data into one or both tone registers, followed by enabling the beeper by setting the BEEP_EN bit in the BEEP_CFG register. The beeper can be started in either pulse mode or sequence mode, depending only on the value of SEQ_REPEAT when BEEP_EN is set. After this bit is enabled, the module begins playing one or more tones. During playback, the module outputs a differential square wave (VSS to VCC) with the frequency and duration based on the BEEP_TONE_A or BEEP_TONE_B register settings. Pulse mode operation completes after playing exactly one tone. Sequence mode operation plays a configurable number of tones before completion. While operating in pulse or sequence mode, the BEEP_BUSY bit is asserted in the BEEP_STAT register; this bit is cleared again upon completion of the output. An interrupt is generated when an event is requested. BEEP_STAT provides feedback on which of the six tracked events has occurred since the register was last cleared; any of these six events can be used for generating interrupts by selecting the corresponding bit in the BEEP_CFG register.

MODES

The beeper has two modes of operation: pulse mode and sequence mode. The operating mode of the beeper is set when the beeper is enabled and cannot be changed until the beeper is disabled. The beeper can be disabled by writing 0x0 to the BEEP_EN bit or by waiting for the beeper to disable itself after the current pulse or sequence completes.

If SEQ_REPEAT has a nonzero value when the beeper is enabled, the beeper operates in sequence mode. If the beeper is enabled with SEQ_REPEAT set to 0x0, the beeper operates in pulse mode.

PULSE MODE

In pulse mode, the beeper plays back exactly one tone. Pulse mode operation begins by enabling the beeper with a zero value in the SEQ_REPEAT field. When enabled, the beeper plays back a single tone as described in the BEEP_TONE_A register. After the playback of this tone has completed, the beeper automatically disables itself, clearing the BEEP_BUSY bit of the BEEP_STAT register.

A currently playing tone can be prematurely terminated by writing 0x0 to the DUR field of the BEEP_TONE_x registers. When a tone is terminated, the appropriate end of the tone event bit is set in the status register and generates an interrupt if selected in the control register. Alternatively, all playback can be immediately terminated by disabling the beeper (clearing the BEEP_EN bit of the BEEP_CFG register); disabling the beeper prevents any events from occurring and, therefore, prevents an interrupt from being generated.

Interrupts are generated as requested in the BEEP_CFG register. Note that the only events that can occur in this mode (and thus be used to generate meaningful interrupts) are for the start and end of BEEP_TONE_A playback.

SEQUENCE MODE

In sequence mode, the beeper plays back a programmable number of tones. Sequence mode operation begins by enabling the beeper with a nonzero value in the SEQ_REPEAT field. After the beeper is enabled, it plays back a series of two-tone sequences as described in the BEEP_TONE_A and BEEP_TONE_B registers in the Tone A Data Register section and the Tone B Data Register section, respectively. The sequences are repeated as many times as is indicated in the SEQ_REPEAT field; a value of 0xFF is a special case value used to run the sequencer indefinitely (or until terminated by user code). After all iterations have been completed, the beeper automatically disables itself, clearing the BEEP_BUSY bit of the BEEP_STAT register.

The number of remaining sequence iterations can be read from the SEQ_REMAIN field of the BEEP_STAT register. When the beeper is running an infinite sequence, SEQ_REMAIN always returns 0xFF. Writing SEQ_REPEAT while the beeper is running in sequence mode restarts the iteration counter to that value and immediately updates the value of the SEQ_REMAIN field.

Sequence mode can be prematurely terminated by writing 0x0 to the SEQ_REPEAT field. Setting SEQ_REPEAT to 0x0 causes the beeper to terminate playback and disable itself after the completion of the current two-tone sequence. When a sequence is terminated, the appropriate end of sequence event bit is set in the status register and generates an interrupt if selected in the control register. Alternatively, all playback can be immediately terminated by disabling the beeper (clearing the BEEP_EN bit of the BEEP_CFG register); disabling the beeper prevents any events from occurring and, therefore, prevents an interrupt from being generated.

Interrupts are generated in sequence mode as requested in the BEEP_CFG register. All beeper events are valid for interrupt generation in this mode.

TONES

The frequency and duration of tones played by the beeper depend on the values stored in the BEEP_TONE_A and BEEP_TONE_B registers. Each of these registers describes a single independent tone. Pulse mode plays tones only from the BEEP_TONE_A register, whereas sequence mode plays tones from both the BEEP_TONE_A and BEEP_TONE_B registers.

The tone registers are broken into three fields: a duration field (DUR), a frequency field (FREQ), and a disable field (DISABLE).

Tone registers can be written at any time. Writing 0x0 to the duration field (DUR) or setting/clearing the disable bit (DISABLE) of the BEEP_TONE_x register for a currently playing tone takes effect immediately. All other modifications to the BEEP_TONE_x registers take effect the next time the tone is played; for most cases, this allows the next tone data to be written to while the current tone is being played.

The duration field is used to program how long a tone plays when selected for playback. Durations are measured in units of 4 ms. Any value from 0 through 255 can be stored in the tone register. A duration of 0x0 causes the tone playback to end immediately. A value of 255 (0xFF) is a special case value used to program a tone for infinite duration; after starting, the tone plays continuously until user code terminates the tone (writes 0x0 to DUR) or disables the beeper (clears the BEEP_EN bit in the BEEP_CFG register).

The frequency field is used to program the relative frequency of the tone with respect to the source clock (32.768 kHz). Writing values 0, 1, 2, and 3 into the frequency field all have the same effect: the output does not oscillate during playback (also known as rest tone). Any value from 4 through 127 (0x7F) can be used to divide the source clock down to the desired tone frequency. This provides a playback range from 8 kHz to ~0.25 kHz.

The disable field is used to provide further control over the output pins of the beeper during playback. When playing a tone with the DISABLE bit set, the output pins behave as though the beeper were disabled; both pins rest at Logic 0 with no dc potential between them and no oscillations. This feature is intended for use when long periods of silence exist in a programmed sequence. It can be used to prevent damage to the piezoelectric component during these periods of silence.

INTERRUPTS AND EVENTS

There are six tracked events that may occur while the beeper is running: Tone A may start or end, Tone B may start or end, and the sequencer may end or be one step away from ending. These six events are always monitored, and when they occur, a sticky bit is set in the BEEP_STAT register to be read by the user at some future time. These bits remain set until cleared by user code.

Any of these six events can also be used to generate an interrupt. Selecting which events generate interrupts is done by setting the respective bits in the BEEP_CFG register. When an event triggers an interrupt, user code must clear the event bit or disable the interrupt selection bit when servicing the interrupt.

When servicing a beeper interrupt, user code must check and eventually clear all event bits in the BEEP_STAT register; multiple events may have triggered the interrupt (if enabled). Writing 0xFF to BEEP_STAT clears all events.

All tracked events are independently selectable for interrupt generation.

CLOCKING AND POWER

The frequency and duration timers of the beeper driver are based on a 32.768 kHz input clock. The clock source is configured system wide, selecting between an external crystal or an internal oscillator by writing to the appropriate clock control module register (CLKCON0: LFCLKMUX). The selected clock provides only a stable reference for the timers; the internal logic of the beeper is clocked by the peripheral clock (PCLK) of the system. For this reason, the beeper cannot run while the system is in a low power mode that gates the peripheral clock (PCLK).

Timer start events are synchronized with the 32 kHz clock. When enabling the beeper, its output may, therefore, be delayed by as much as one 32 kHz clock period (30,520 ns). For a 4 MHz PCLK, this results in up to 122 PCLK periods between starting the beeper and actually producing audio output. Any changes to the BEEP_TONE_x registers during this time affects the pending audio. User code must ensure that the tone is playing before modifying the tone registers either by polling the status register or by setting an interrupt to assert on the start of tone events.

User code must disable the beeper prior to entering a low power state where the peripheral clock is gated. Damage to the piezoelectric component may occur if the system enters a low power mode while the beeper is playing a tone due to constant, long-term dc potential across the terminals of the piezoelectric component.

POWER-DOWN CONSIDERATIONS

There is the possibility of irreversible damage to the external piezoelectric beeper device if the beeper is enabled and driving the TONE_P and TONE_N outputs under the following conditions:

- Entering hibernate power mode
- Entering SYS_SLEEP power mode
- Turning off PCLK

The pins do not automatically disengage from the beeper module in these modes.

TIMING DIAGRAM

For illustration, the timing diagram in Figure 80 shows the behavior of the beeper when programmed to generate a tone. In this example, the tone is first written to the BEEP_TONE_A register, followed by enabling the beeper by writing to the BEEP_CFG register. Within this figure, t_{DEL} and t_{FREQ} denote the following:

- t_{DEL} illustrates the maximum one 32 kHz clock period delay between enabling the beeper and the actual start of playback.
- t_{FREQ} illustrates the output period of the two differential pins; for a 1 kHz (1024 Hz) tone, t_{FREQ} = 0.9766 ms.

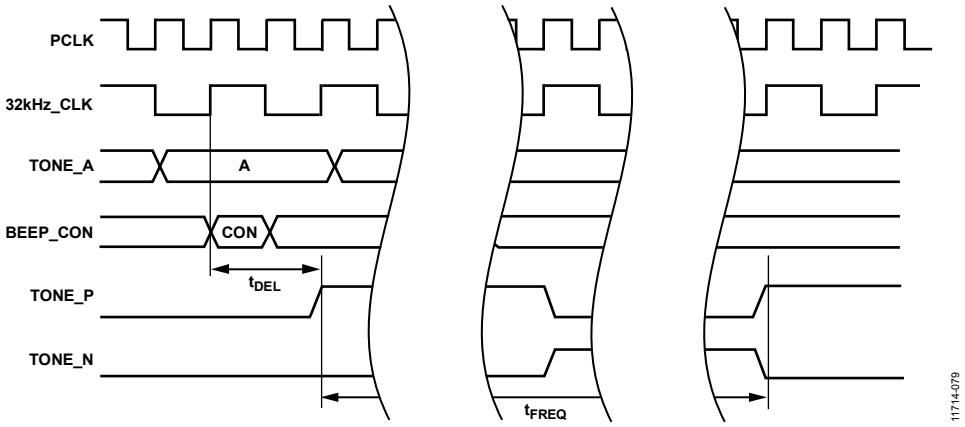


Figure 80. Timing Diagram

PROGRAMMING EXAMPLES

Example 1

The beeper driver is programmed for a single 1 kHz beep of 1 sec length with an interrupt when the tone is done playing.

1. Set Tone A duration to 1 ms. BEEP_TONE_A: DUR = 0xFA; // (250 × 4 ms).
2. Set Tone A frequency to 1 kHz. BEEP_TONE_A: FREQ = 0x20; // (32 kHz/32).
3. Interrupt at end of Tone A. BEEP_CFG: IRQ_TONEA_END = 0x1.
4. Set the sequence repeat to 0. BEEP_CFG: SEQ_REPEAT = 0x0.
5. Enable the beeper. BEEP_CFG: BEEP_EN = 0x1.

The register access sequence is BEEP_TONE_A = 0x20FA and BEEP_CFG = 0x0900.

Example 2

The beeper driver is programmed for a melody of 32 two-tone sequences of the note G# for 500 ms and F# for 1000 ms with an interrupt at the end of the melody.

1. Set Tone A duration to 500 ms. BEEP_TONE_A: DUR = 0x7D; // (125 × 4 ms).
2. Set Tone A frequency to G#. BEEP_TONE_A: FREQ = 0x27; // (32 kHz/39 = 840 Hz).
3. Set Tone B duration to 1000 ms. BEEP_TONE_B: DUR = 0xFA; // (250 × 4 ms).
4. Set Tone B frequency to F#. TONEB: FREQ = 0x2C; //(32 kHz/44 = 745 Hz).
5. Set interrupt at end of sequence. BEEP_CFG: IRQ_SEQ_END = 0x1.
6. Set sequence repeat to 32. BEEP_CFG: SEQ_REPEAT = 0x20.
7. Enable the beeper. BEEP_CFG: BEEP_EN = 0x1.

The register access sequence is BEEP_TONE_A = 0x277D, BEEP_TONE_B = 0x2CFA, and BEEP_CFG = 0x8120.

BEEPER DRIVER MEMORY MAPPED REGISTERS

Beeper Driver Register Map

Table 432. Beeper Driver Register Summary

Address	Name	Description	Reset	RW
0x40005C00	BEEP_CFG	Beeper configuration register	0x0000	RW
0x40005C04	BEEP_STAT	Beeper status register	0x0000	RW
0x40005C08	BEEP_TONE_A	Tone A data register	0x0001	RW
0x40005C0C	BEEP_TONE_B	Tone B data register	0x0001	RW

Beeper Configuration Register

Address: 0x40005C00, Reset: 0x0000, Name: BEEP_CFG

Table 433. Bit Descriptions for BEEP_CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ_SEQ_END	Sequence end IRQ. Set this bit to request an interrupt on the event the sequencer has stopped playing. 0: IRQ not generated at end of sequence. 1: IRQ generated at end of sequence.	0x0	RW
14	IRQ_SEQ_NEAR_END	Sequence 1 cycle from end IRQ. Set this bit to request an interrupt on the event the sequencer is currently running and has just one repetition remaining before completion. 0: IRQ not generated one tone pair before sequence ends. 1: IRQ generated one tone pair before sequence ends.	0x0	RW
13	IRQ_TONEB_END	Tone B end IRQ. Set this bit to request an interrupt on the event Tone B stops playing. 0: IRQ not generated at end of Tone B. 1: IRQ generated at end of Tone B.	0x0	RW
12	IRQ_TONEB_START	Tone B start IRQ. Set this bit to request an interrupt on the event Tone B starts playing. 0: IRQ not generated at start of Tone B. 1: IRQ generated at start of Tone B.	0x0	RW
11	IRQ_TONEA_END	Tone A end IRQ. Set this bit to request an interrupt on the event Tone A stops playing. 0: IRQ not generated at end of Tone A. 1: IRQ generated at end of Tone A.	0x0	RW
10	IRQ_TONEA_START	Tone A start IRQ. Set this bit to request an interrupt on the event Tone A starts playing. 0: IRQ not generated at start of Tone A. 1: IRQ generated at start of Tone A.	0x0	RW
9	RESERVED	Reserved.	0x0	RESERVED
8	BEEP_EN	Beeper enable. Write 0x1 to this bit to start playing Tone A. This bit plays a single tone if SEQ_REPEAT is zero, otherwise it plays a series of two-tone sequences. Write 0x0 to this bit at any time to end audio playback. Reading this bit always returns 0x0, read BEEP_BUSY from the BEEP_STAT register to determine if the beeper is currently enabled. 0: enable module. 1: disable module.	0x0	RWAC
[7:0]	SEQ_REPEAT	Beeper sequence repeat value. When the beeper transitions to an enabled state, the value of this field selects whether the beeper runs in pulse mode or sequence mode. If 0x0, the beeper runs in pulse mode and plays back only one tone (Tone A) before being disabled. If a nonzero value is written to this field, the beeper runs in sequence mode and plays the two-tone sequence (Tone A, Tone B) the requested number of times before being disabled. Updates to this field have no affect while running in pulse mode. Updates to this field take immediate affect when running in sequence mode. 0x0: enabling starts in pulse mode. 0x1 to 0xFE: enabling starts in sequence mode and play a finite number of notes. 0xFF: enabling starts in sequence mode and play forever until stopped by user code.	0x0	RW

Beeper Status Register

Address: 0x40005C04, Reset: 0x0000, Name: BEEP_STAT

Table 434. Bit Descriptions for BEEP_STAT

Bits	Bit Name	Description	Reset	Access
15	EVT_SEQ_END	Sequencer has ended. This bit is asserted whenever the beeper stops running in sequence mode. It is cleared only by user code writing 0x1 to this location.	0x0	RW1C
14	EVT_SEQ_NEAR_END	Sequencer last tone pair has started. This bit is asserted whenever the beeper is running in sequence mode and has only one iteration of sequences left to play. It is cleared only by user code writing 0x1 to this location.	0x0	RW1C
13	EVT_TONEB_END	Tone B has ended. This bit is asserted whenever the beeper stops playing Tone B. It is cleared only by user code writing 0x1 to this location.	0x0	RW1C
12	EVT_TONEB_START	Tone B has started. This bit is asserted whenever the beeper begins playing Tone B. It is cleared only by user code writing 0x1 to this location.	0x0	RW1C
11	EVT_TONEA_END	Tone A has ended. This bit is asserted whenever the beeper stops playing Tone A. It is cleared only by user code writing 0x1 to this location.	0x0	RW1C
10	EVT_TONEA_START	Tone A has started. This bit is asserted whenever the beeper begins playing Tone A. It is cleared only by user code writing 0x1 to this location.	0x0	RW1C
9	RESERVED	Reserved.	0x0	RESERVED
8	BEEP_BUSY	Beeper is busy. This bit is asserted after enabling the beeper module by writing 0x1 to BEEP_EN in the BEEP_CFG register. The bit is cleared when the module completes its operation or when user code writes 0x0 to BEEP_EN. 0: beeper is currently disabled. 1: beeper is currently enabled.	0x0	R
[7:0]	SEQ_REMAIN	Remaining tone pair sequence iterations to play in sequence mode. After a sequence has ended, this field resets to SEQ_REPEAT. This field is updated as the beeper plays back audio in sequence mode. Each two-tone iteration starts by playing Tone A and ends by playing Tone B. This field is updated at the conclusion of Tone B playback and therefore during the last iteration returns 0x1 during Tone A and Tone B playback. When playing an infinite sequence this field always returns 0xFF.	0x0	R

Tone A Data Register

Address: 0x40005C08, Reset: 0x0001, Name: BEEP_TONE_A

Tone A is the first tone to play in sequence mode, and the only tone to play in pulse mode. Writing 0x0 to the DUR field while Tone A is playing immediately terminates the tone. All other writes to BEEP_TONE_A affect the next play back of the tone only and do not affect the currently playing tone.

Table 435. Bit Descriptions for BEEP_TONE_A

Bits	Bit Name	Description	Reset	Access
15	DISABLE	Output disable. This bit is set to disable the beeper output while Tone A is playing. The beeper holds both of its two output pins at Logic 0 when disabled. Writes to this bit take effect immediately if Tone A is currently playing. 0: output enabled. 1: output disabled, no dc potential across output pins.	0x0	RW
[14:8]	FREQ	Tone frequency. This field defines the frequency for Tone A as integer divisions of the 32.768 kHz clock. The values 0 through 3 are interpreted as rest tone and results in no oscillations of the beeper output pins. All other values are directly used to divide the 32 kHz input clock. Writes to this field immediately affect the output of Tone A if currently playing. 0 to 3: rest tone (duration but no oscillation). 4 to 127: oscillation at 32 kHz/(FREQ).	0x0	RW
[7:0]	DUR	Tone duration. This field defines the duration for Tone A in 4 ms increments. Writing a value of 0x0 immediately terminates Tone A if it is currently playing. Writing a value of 0xFF causes Tone A to play forever once started, or until user code terminates the tone. Only a write of 0x0 affects a currently playing tone, all other values written are used only the next time Tone A is played. 0 to 254: tone plays for (DUR) × 4 ms period. 255: tone plays for infinite duration.	0x1	RW

Tone B Data Register

Address: 0x40005C0C, Reset: 0x0001, Name: BEEP_TONE_B

Tone B is the second tone to play in sequence mode, and is not played pulse mode. Writing 0x0 to the DUR field while Tone B is playing immediately terminates the tone. All other writes to BEEP_TONE_B affect the next play back of the tone only and do not affect the currently playing tone.

Table 436. Bit Descriptions for BEEP_TONE_B

Bits	Bit Name	Description	Reset	Access
15	DISABLE	Output disable. This bit is set to disable the beeper output while Tone B is playing. The beeper holds both of its two output pins at Logic 0 when disabled. Writes to this bit take effect immediately if Tone B is currently playing. 0: output enabled. 1: output disabled, no dc potential across output pins.	0x0	RW
[14:8]	FREQ	Tone frequency. This field defines the frequency for Tone B as integer divisions of the 32.768 kHz clock. The values 0 through 3 are interpreted as rest tone and result in no oscillations of the beeper output pins. All other values are directly used to divide the 32 kHz input clock. Writes to this field immediately affect the output of Tone B if currently playing. 0 to 3: rest tone (duration but no oscillation). 4 to 127: oscillation at 32 kHz/(FREQ).	0x0	RW
[7:0]	DUR	Tone duration. This field defines the duration for Tone B in 4 ms increments. Writing a value of 0x0 immediately terminates Tone B if currently playing. Writing a value of 0xFF causes Tone B to play forever after starting, or until user code terminates the tone. Only a write of 0x0 affects a currently playing tone, all other values written are used only the next time Tone B is played. 0 to 254: tone plays for (DUR) × 4 ms period. 255: tone plays for infinite duration.	0x1	RW

LCD CONTROLLER

The display interface of the ADuCM350 contains an on-chip LCD controller capable of directly driving an LCD panel. A total of 36 pins are available on the ADuCM350 for LCD functionality. The LCD controller is functional in all power-down modes except hibernate mode.

FEATURES

Features of the LCD controller include the following:

- Supports driving up to 128 segments
- Selectable multiplex options; static and 4× multiplexing
- Static: 1 backplane × 32 frontplanes
- 4× mux: 4 backplanes × 32 frontplanes
- The LCD waveform voltages are generated using an internal charge pump circuitry and can support bias levels from 2.4 V up to 3.6 V LCDs
- Programmable frame rates
- Interrupt generation at frame boundary for updating LCD data
- Information for up to two screens can be stored separately
- LCD frame clock generated by using the on-board 32 kHz crystal
- Extra segment lines that are not connected to the LCD cannot be used as GPIOs

LCD SOFTWARE SETUP

The LCD configuration register (LCDCON) configures the LCD module to drive the type of LCD in the user end system. The LCDMUX bit in this register must be set according to the specifications of the LCD used in the application. Table 437 indicates supported combinations of LCDMUX and bias values.

LCD CHARGE PUMP AND EXTERNAL CAPACITOR REQUIREMENTS

The LCD waveform voltages are generated using an internal charge pump circuitry. The 32 kHz regulated buck boost device can support bias levels from 2.4 V up to 3.6 V LCDs with a minimum battery voltage of 2.1 V.

A ratio of LCD segment capacitance to reservoir capacitor (VLCDVDD pin to LCDGND pin) can be determined as 10,000:1. A typical value of 1 µF covers a segment capacitance of up to 100 pF.

VLCD_FLY1 and VLCD_FLY2 can be left as a no connect.

When the segment driver is not used, connect the VLCDVDD pin to the VCCM supply of the VCCM pin.

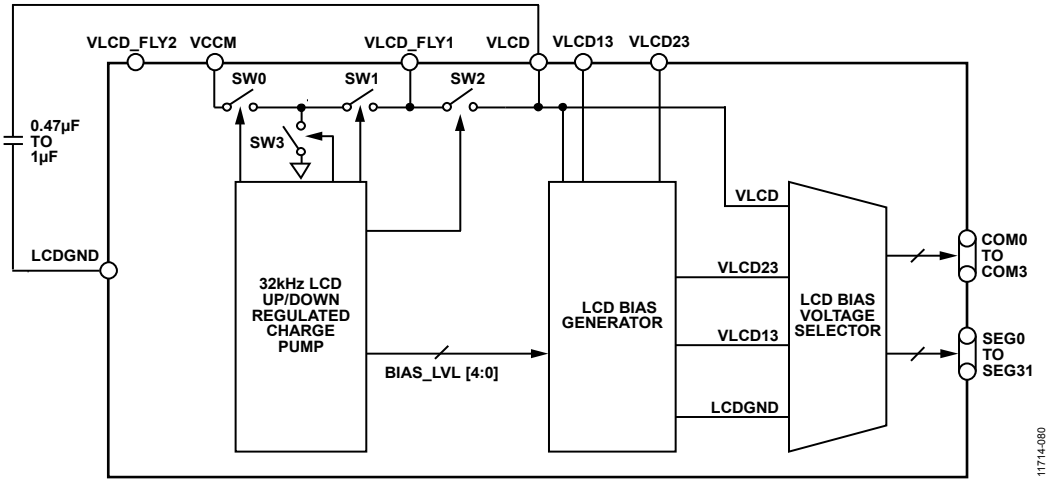


Figure 81. LCD Driver Block Diagram

LCD MULTIPLEX TYPES

The LCD controller can be configured to support the following two multiplex types by programming the LCDMUX bit in the LCDCON register:

- Static (only COM0 is used)
- 4× multiplex (COM0 to COM3 are used)

LCD BIAS TYPES

The LCD controller supports the following two bias types, depending on the LCDMUX bit in the LCDCON register.

- Static bias (two voltage levels: V1 and V4)
- 1/3 bias (four voltage levels: V1, V2, V3, and V4)

Table 437. Legal Combinations of LCDMUX and BIASLVL Bits

Multiplex Type	1/1 Bias	1/3 Bias
Static	Yes	No
4× Multiplex	No	Yes

The Display Element Control section contains details about setting up the LCD data memory to turn individual LCD segments on and off. Setting the LCDRST bit in the LCD configuration register (LCDCON) resets the LCD data memory to its default (0). A power-on reset also clears the LCD data memory.

LCD SCREEN SELECTION

The LCD controller implements two sets of data registers. Therefore, the ADuCM350 can store two separate screens in its data registers. By default, Screen 0 is selected, and the corresponding segment information needs to be written to the LCDDATAn_S0 registers, where n = 0 to 7. The SCREENSEL bit in LCDCON can be used to select between the two screens. The segment information for Screen 1 must be written to registers LCDDATAn_S1, where n = 0 to 7.

LCD BIAS VOLTAGE

The bias level is controlled using BIASLVL (Bits[4:0] of the LCDCONTRAST register). The voltage step for the bias level is 38.7 mV (approximately 40 mV). The bias voltage for a given BIASLVL value is given by the following equation:

$$Bias\ Voltage = 2.4\ V + (0.0387) \times BIASLVL$$

Table 438. LCD Bias Level Control

BIASLVL	Bias Voltage (V)
00000	2.4
00001	2.4387
…	…
…	…
11110	3.5613
11111	3.6

LCD TIMING AND WAVEFORMS

An LCD segment acts like a capacitor that is charged and discharged at a certain rate. This rate, called the refresh rate, determines the visual characteristics of the LCD. A slow refresh rate results in flickering with the LCD blinking on and off between refreshes and is disconcerting to the user. A fast refresh rate results in ghosting, where the LCD screen appears to be continuously lit due to the segment not turning off. In addition, a faster refresh rate consumes more power.

The frame rate, or refresh rate, for the LCD controller is the number of times an LCD segment is energized per second, which is derived from the external 32 kHz crystal oscillator, f_{LCDCLK}. The crystal oscillator can be turned on by a register bit for all power modes, except hibernate mode. The minimum refresh rate needed for the LCD to appear solid (without blinking) is independent of the multiplex level.

The LCD waveform frequency, f_{LCD}, is the frequency at which the LCD switches the common and segment line. This frequency depends heavily on the multiplex level. f_{LCD} and the frame rate are set by f_{LCDCLK}, the multiplex level, and the frame rate selection bits, FRAMESEL, in the LCDCON register.

The f_{LCDCLK} (32 kHz) is prescaled differently for different multiplex schemes to generate a base clock frequency f_{BCLK}, as shown in Figure 82. The base frequency for a static multiplex is 1 kHz and for a 4× multiplex is 4 kHz. These frequencies are chosen to provide a number of frame rate options in the 30 Hz to 80 Hz range. The frequency f_{LCD} is given by $f_{LCD} = f_{BCLK}/(FRAMESEL + 4)$.

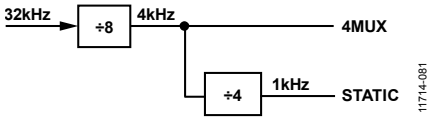


Figure 82. LCD Controller Prescaler

The LCD controller provides 16 selectable values for f_{LCD}; therefore, 16 values can be selected for the frame rate as well. The frame rate is set by the FRAMESEL bits in the LCDCON register (see Table 439).

Each frame of the LCD waveform has a noninverted and an inverted component. Therefore, the maximum frame rate is half the LCD waveform frequency ($f_{LCD}/2$), and each frame has an average dc offset of 0. This is necessary because a dc offset degrades the lifetime and performance of the LCD.

The FRAMEINV bit in the LCDCON register allows frames to be inverted every alternate frame. If this mode is selected, the dc cancellation is done twice over two frames (once within each frame and once frame to frame). This is helpful to eliminate the dc offset that occurs due to the difference in rise and fall times of the COM/SEG lines. This bit is only for multiplex mode.

In static mode, frame inversion always happens, even if FRAMEINV is set to 0. The outputs from tests with FRAMEINV set or clear are identical.

Table 439. Bit Descriptions for FRAMESEL (Bits [7:4] of the LCDCON Register)

				Static Mux		4× Mux	
FRAMESEL[3]	**FRAMESEL[2]**	**FRAMESEL[1]**	**FRAMESEL[0]**	**f_{LCD} (Hz)**	**Frame Rate (Hz)**	**f_{LCD} (Hz)**	**Frame Rate (Hz)**
0	0	0	0	256	128	1024	128
0	0	0	1	204.8	102.4	819.2	102.4
0	0	1	0	170.7	85.3	682.7	85.3
0	0	1	1	146.3	73.1	585.1	73.1
0	1	0	0	128	64	512	64
0	1	0	1	113.8	56.9	455.1	56.9
0	1	1	0	102.4	51.2	409.6	51.2
0	1	1	1	93.1	46.5	372.4	46.5
1	0	0	0	85.3	42.7	341.3	42.7
1	0	0	1	78.8	39.4	315.1	39.4
1	0	1	0	73.1	36.6	292.6	36.6
1	0	1	1	68.3	34.1	273.1	34.1
1	1	0	0	64	32	256	32
1	1	0	1	60.2	30.1	240.9	30.1
1	1	1	0	56.9	28.4	227.6	28.4
1	1	1	1	53.9	26.9	215.6	26.9

BLINK MODE

Blink mode is enabled by setting the BLINKEN bit in the LCDCON register. There are two ways in which this mode can be used that include the following:

- When the BLINKEN bit in the LCDCON register is set and the AUTOSWITCH bit in LCDBLINK is not set, the LCD controller alternates between the LCD on state and the LCD off state so that the LCD screen appears to blink. The screen that is turned on and off is selected using the SCREENSEL bit. There are two blink modes: a software controlled blink mode and an automatic blink mode.
- When the BLINKEN and the AUTOSWITCH bits are both set, the LCD controller alternately displays information from Screen 0 and Screen 1.
- When any write to the LCDBLINK register occurs, the blink counters are all reset and begin counting again from 0. For example, if the BLKFREQ is unchanged and AUTOSWITCH is enabled or disabled, the current pulse is curtailed.

Software Controlled Blink Mode

The LCD blink rate can be controlled by code running on the Cortex-M3 by toggling the BLKMOD bits in the LCDBLINK register. The display is switched between on and off states, or between Screen 0 and Screen 1, at a rate that is determined by the processor code.

Automatic Blink Mode

Blink rate can be automatically controlled using hardware. Up to five blink rates can be selected by the BLKMOD bits and the BLKFREQ bits in the LCDBLINK register. Blink rates of 4 Hz, 2 Hz, 1 Hz, 2/3 Hz, and 1/2 Hz are supported.

Screen Switch Mode

The ADuCM350 can store up to two screens in its data registers. Switching between both the two screens can be selected by setting the AUTOSWITCH bit in the LCDBLINK register. The frequency with which the switching happens is selected by the BLKMOD and BLKFREQ bits in the LCDBLINK register. Blink rates of 4 Hz, 2 Hz, 1 Hz, 2/3 Hz, and 1/2 Hz are supported.

OPERATION DURING ADuCM350 POWER MODES

The LCD controller continues to operate when the ADuCM350 is in CORE_SLEEP and SYS_SLEEP power modes. In both of these modes, although the LCD data cannot be changed, the controller continues to display the contents of the selected screen. Blink operation can also continue either using the automatic hardware blink mode or the screen switch mode. In only power gated hibernate mode, the LCD controller does not drive the LCD. Before entering hibernate mode, the LCD controller must be disabled by clearing the LCDEN bit of LCDCON. When the ADuCM350 exits hibernate mode, it must follow the initialization sequence discussed in the Contrast Control section. It must also program the new frame to be displayed in the LCDDATAn_Sx registers. Only then, the LCD controller must be reenabled. This ensures that the LCD does not display any stale data (which was displayed before entering the hibernate mode).

DISPLAY ELEMENT CONTROL

The on or off state of each segment of the LCD is controlled by individual register bits in two banks of eight data registers. The data registers for Screen 0 are LCDDATAn_S0 and for Screen 1 are LCDDATAn_S1, where n is 0 to 7. Each register configures the on and off states of the segment lines depending on the multiplex level. For 4× multiplexing, frame data is organized as a set of four common lines per segment line. For example, the first four bits of LCDDATA0_S0 refer to Segment Line 0 values for all the common lines. The next four bits refer to Segment Line 1 values for all the common lines. Therefore, if a smaller display panel were chosen, where the number of segment lines is fewer, although common lines are still 4, the user can only use registers up to the maximum segment number.

Table 440 shows the correlation of each bit in the LCDDATA registers to the respective backplane and frontplane signals for 4× multiplexing with four backplanes and 32 frontplanes.

Table 440. 4 Backplanes (BP) and 32 Frontplanes (FP)

LCD Data Addr	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	BP 3	BP 2	BP 1	BP 0	BP 3	BP 2	BP 1	BP 0	BP 3	BP 2	BP 1	BP 0	BP 3	BP 2	BP 1	BP 0
0x1C	FP31	FP31	FP31	FP31	FP30	FP30	FP30	FP30	FP29	FP29	FP29	FP29	FP28	FP28	FP28	FP28
0x18	FP27	FP27	FP27	FP27	FP26	FP26	FP26	FP26	FP25	FP25	FP25	FP25	FP24	FP24	FP24	FP24
0x14	FP23	FP23	FP23	FP23	FP22	FP22	FP22	FP22	FP21	FP21	FP21	FP21	FP20	FP20	FP20	FP20
0x10	FP19	FP19	FP19	FP19	FP18	FP18	FP18	FP18	FP17	FP17	FP17	FP17	FP16	FP16	FP16	FP16
0x0C	FP15	FP15	FP15	FP15	FP14	FP14	FP14	FP14	FP13	FP13	FP13	FP13	FP12	FP12	FP12	FP12
0x08	FP11	FP11	FP11	FP11	FP10	FP10	FP10	FP10	FP9	FP9	FP9	FP9	FP8	FP8	FP8	FP8
0x04	FP7	FP7	FP7	FP7	FP6	FP6	FP6	FP6	FP5	FP5	FP5	FP5	FP4	FP4	FP4	FP4
0x00	FP3	FP3	FP3	FP3	FP2	FP2	FP2	FP2	FP1	FP1	FP1	FP1	FP0	FP0	FP0	FP0

Table 441 shows the static multiplexing with 1 backplane and 32 frontplanes.

Table 441. 1 Backplane and 32 Frontplanes[1]

LCD Data Addr	B15 BP 0	B14 BP 0	B13 BP 0	B12 BP 0	B11 BP 0	B10 BP 0	B9 BP 0	B8 BP 0	B7 BP 0	B6 BP 0	B5 BP 0	B4 BP 0	B3 BP 0	B2 BP 0	B1 BP 0	B0 BP 0
0x1C	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0x18	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0x14	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0x10	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0x0C	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0x08	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0x04	FP31	FP30	FP29	FP28	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20	FP19	FP18	FP17	FP16
0x00	FP15	FP14	FP13	FP12	FP11	FP10	FP8	FP8	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0

[1] NA means not applicable.

Writing to the LCD Registers

Before enabling the LCD, all the screen data has to be written by the core. After the display process has started, new frame data must be written only after the FRAMEINT is received. The LCD frame boundary interrupt can be used to synchronize writing to the data registers before the start of a new frame. Writing the data registers at the frame boundary allows a crisp transition of the image displayed on the LCD. The FRAMEINT_EN bit in the LCDCON register can be used to enable the LCD controller to interrupt the processor. When this bit is set, an interrupt is generated by the LCD controller just before the frame boundary (after the data registers are accessed to display the last set of segments within a given frame). The processor can then write the LCD data registers with new display data. This new data must be completed before the frame boundary for a flicker free transition. After the data is written, the FRAMEINT bit in the LCDSTAT register must be cleared by writing a 1 to the bit. The frame data write is easily completed within this time window because the core has a minimum of 244 APB clock cycles (for the worst case of SYS_CLK = 253 kHz and the maximum LCD frame rate of 128 Hz). Because only eight 16-bit registers need to be written for one screen, the user can assign the highest priority to the LCD frame interrupt, quickly service the interrupt, and then proceed to servicing other interrupts. Therefore, there are not any issues in the frame display transition.

For the core to write some data when the frame interrupt is not enabled, it must check if the SAFE_TO_WR bit in the LCDSTAT register is asserted. Otherwise, it must wait until this bit is set. Alternatively, it can enable the frame interrupt and then write the data when the interrupt occurs.

All other control register updates are also synchronized to the frame boundary. Therefore, any LCD configuration registers updates must also occur only when SAFE_TO_WR is asserted.

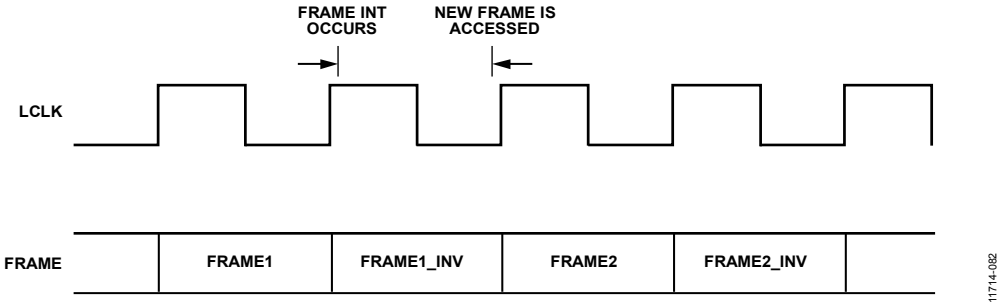


Figure 83. Frame Interrupt and Register Update

Contrast Control

The LCD bias voltage sets the contrast of the display using the charge pump provided. The ADuCM350 provides 32 bias levels selected by the BIASLVL bits in the LCDCONTRAST register.

On power-up, the core must set the CP_EN bit in the LCDCONTRAST register to enable the charge pump. At the same time, the CP_PD bit in the LCDCONTRAST register must be cleared. The core must initiate a timeout for 62.5 ms using any of the general-purpose timers (GPTs). After receiving the timeout interrupt, the core must check the CP_GD status in the LCDSTAT register. If this bit is low, there is an error in charge pump or the LCD glass. In this case, nothing can be displayed on the LCD. If this bit is high after the timeout, then the core must set the CPINT_EN bit in the LCDCON register to enable an interrupt whenever CP_GD goes low. Whenever there is a CP_GD interrupt, the core must clear the CPINT_EN bit and again wait for a 62.5 ms timeout. If the CP_GD bit is still low, then there is a fault condition. Because there may be glitches in the CP_GD signal when exiting hibernate mode, this interrupt must be disabled when the device enters hibernate mode. The timeout interrupt enable sequence must be repeated whenever the ADuCM350 exits hibernate mode.

Similarly, as soon as the core programs the BIASLVL bits, it must initiate a timeout for 62.5 ms using any of the GPTs. After receiving the timeout interrupt, the core must check for the VLCD_OK status in the LCDSTAT register. If this bit is not set, it indicates that the required bias level was not reached in the charge pump, which is a fault condition. Repeat this sequence each time the BIASLVL value must be changed.

Like power-up, whenever the ADuCM350 exits hibernate mode, the LCDCONTRAST register must be reprogrammed, and the charge pump status must be checked after the timeout.

LCD MEMORY MAPPED REGISTERS

LCD Register Map

Table 442. LCD Register Summary

Address	Name	Description	Reset	RW
0x40008000	LCDCON	LCD configuration register	0x0002	RW
0x40008004	LCDSTAT	LCD status register	0x0012	RW
0x40008008	LCDBLINK	LCD blink control register	0x0000	RW
0x4000800C	LCDCONTRAST	LCD contrast control register	0x0040	RW
0x40008010	LCDDATAn_S0	Screen 0 LCD Data Register n	0x0000	RW
0x40008030	LCDDATAn_S1	Screen 1 LCD Data Register n	0x0000	RW

LCD Configuration Register

Address: 0x40008000, Reset: 0x0002, Name: LCDCON

Table 443. Bit Descriptions for LCDCON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	RESERVED
12	BLINKEN	Blink mode enable. 0: blink mode not enabled. 1: blink mode is enabled. This is configured by the BLKMOD and BLKFREQ bits in the LCDBLINK register.	0x0	RW
11	CPINT_EN	Enable charge pump interrupt. 0: no interrupt is generated when CP_GD goes low. 1: an interrupt is generated when CP_GD goes low.	0x0	RW
10	FRAMEINT_EN	Enable frame boundary interrupt. 0: do not enable interrupt frame boundary interrupt. 1: enable interrupt when frame boundary is reached.	0x0	RW
9	LCDRST	LCD data registers reset. 0: LCD data registers are not reset. 1: LCD data registers are reset to 0.	0x0	RWAC
8	FRAMEINV	Frame inversion mode enable. 0: frames are not inverted. 1: frames are inverted at each frame boundary.	0x0	RW
[7:4]	FRAMESEL	LCD frame rate select.	0x0	RW

Bits	Bit Name	Description	Reset	Access
3	SCREENSEL	Screen select. This bit selects the screen that is output on the LCD pins. 0: select Screen 0. 1: select Screen 1.	0x0	RW
1	LCDMUX	LCD multiplex value. 0: static (COM0). 1: 4× multiplexing (COM3:COM0).	0x1	RW
0	LCDEN	LCD enable. 0: LCD controller not enabled. 1: LCD controller enabled.	0x0	RW

LCD Status Register

Address: 0x40008004, Reset: 0x0012, Name: LCDSTAT

Table 444. Bit Descriptions for LCDSTAT

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	RESERVED
4	LCD_IDLE	LCD idle state. This bit can be used to check if the LCD is actively displaying or not. If the LCDEN bit is cleared, this bit must be read to know when the current frame refresh cycle ends. 0: LCD is actively displaying. 1: LCD is idle.	0x1	R
3	VLCD_OK	VLCD update complete. 0: VLCD on reservoir capacitor. is not equal to BIASLVL. 1: VLCD on reservoir capacitor. is equal to BIASLVL.	0x0	R
2	CP_GD	Charge pump good. This bit reflects the status of the charge pump. If the CPINT_EN bit in LCDCON is cleared, this bit is not sticky. If CPINT_EN is set, then an interrupt is generated whenever CP_GD becomes low and this bit becomes sticky. The low value is retained until CPINT_EN is cleared. 0: VLCD on reservoir capacitor is < 2.1 V. 1: VLCD on reservoir capacitor is ≥2.1 V.	0x0	R
1	SAFE_TO_WR	Safe to write the registers. The new frame data must be updated only if this bit is set. Note that if this bit is 0, wait for the next frame interrupt to update the screen/control registers. 0: LCD registers must not be updated. 1: LCD registers can be safely updated.	0x1	R
0	FRAMEINT	LCD frame boundary interrupt. 0: frame boundary interrupt not set. 1: frame boundary interrupt set. When set, this can be reset by writing a 1 to this bit.	0x0	RW1C

LCD Blink Control Register

Address: 0x40008008, Reset: 0x0000, Name: LCDBLINK

Table 445. Bit Descriptions for LCDBLINK

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved.	0x0	RESERVED
5	AUTOSWITCH	Switch screen automatically. If BLINKEN is cleared, this bit has no effect. 0: if BLINKEN is set, LCD switches between on and off states of the screen selected using the SCREENSEL bit. 1: if BLINKEN is set, LCD switches between Screen 0 and Screen 1.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[4:2]	BLKFREQ	Blink rate configuration bits. This is the frequency at which the LCD switches screens. If the blink frequency is set as 4 Hz, the LCD displays one screen for 125 ms and the alternate screen for 125 ms. 000: 4 Hz. 001: 2 Hz. 010: 1 Hz. 011: 2/3 Hz. 100: 1/2 Hz. 101: reserved. 110: reserved. 111: reserved.	0x0	RW
[1:0]	BLKMOD	Blink mode clock source configuration bits. For all blink modes (software or hardware), switching between screens (0 and 1) or on/off state of the selected screen is controlled using the AUTOSWITCH bit. 00: blink rate controlled by software; display is off or the alternate screen is displayed. 01: blink rate controlled by software; display is on and the selected screen is displayed. 10: reserved. 11: blink rate controlled by hardware; blink rate is set by the BLKFREQ bits.	0x0	RW

LCD Contrast Control Register

Address: 0x4000800C, Reset: 0x0040, Name: LCDCONTRAST

Table 446. Bit Descriptions for LCDCONTRAST

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	RESERVED
6	CP_PD	Charge pump power down. 0: charge pump is not powered down. 1: charge pump is powered down.	0x1	RW
5	CP_EN	Charge pump enable. 0: charge pump is not enabled. 1: charge pump is enabled.	0x0	RW
[4:0]	BIASLVL	Bias level selection. See Table 438 for more details.	0x0	RW

Screen 0 LCD Data Register (0 to 7)

Address: 0x40008010 to 0x4000802C (Increments of 0x4), Reset: 0x0000, Name: LCDDATAn_S0

Table 447. Bit Descriptions for LCDDATAn_S0

Bits	Bit Name	Description	Reset	Access
[16:0]	LCDDATAn_S0	LCD data for turning segments on and off.	0x0000	RW

Screen 1 LCD Data Register (0 to 7)

Address: 0x40008030 to 0x4000804C (Increments of 0x4), Reset: 0x0000, Name: LCDDATAn_S1

Table 448. Bit Descriptions for LCDDATAn_S1

Bits	Bit Name	Description	Reset	Access
[16:0]	LCDDATAn_S1	LCD data for turning segments on and off.	0x0000	RW

PARALLEL DISPLAY INTERFACE (PDI)

The on-chip LCD controller is capable of driving an external LCD display module. It has a total of 25 pins for the display interface, which can support data transfers up to 16 bits.

FEATURES

Features include the following:

- Supports Type A, Type B, and Type C of the MIPI DBI specification, Version 2.00.
- Supports both Fixed E mode and Clocked E mode options of Type A interface.
- Supports all the bus width options (8-/9-/16-bit data) for Type A and Type B.
- Supports 9-bit (Option 1) and 8-bit (Option 3) serial interface for Type C.
- Supports TE (tearing effect) signal.
- Supports a video frame rate of TBD fps (frames per second) for a 240 × 430 resolution LCD panel.
- Depth on 8-bit interface is 8 bits/12 bits/16 bits per pixel (not 18 bits or 24 bits).
- Depth on 9-bit interface is 18 bits per pixel (not 8 bits, 12 bits, 16 bits, or 24 bits).
- Depth on 16-bit interface is 8 bits/12 bits/16 bits per pixel (not 18 bits or 24 bits).
- Interrupt generation at frame boundary (using TE signal) for updating LCD data.
- DMA request generation capability for frame data.
- Programmable t_{CYCLE} values with a minimum value of 310 ns (minimum granularity of 62 ns for a core clock frequency of 16 MHz).

NOT SUPPORTED

Features not supported include the following:

- 16-bit (Option 2) serial interface of DBI Type C.
- 24-bits per pixel format (hardware has no restrictions to support this).

BYTE PACKING OF PIXEL DATA

A 3 bits per pixel (BPP) format needs to be supported for Type 2 or Type 3 display architectures. These architectures can be used for low cost LCD panels. If so, a 3 BPP format can also be supported by the core. Data can be organized in the memory using one of the two options shown Table 449 and Table 450.

Table 449. 3 BPP Option 1 (X = Don't Care)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	R1	G1	B1	R2	G2	B2

Table 450. 3 BPP Option 2 (X = Don't Care)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	R1	G1	B1	X	R2	G2	B2

The 8-bit data can be packed on byte boundaries. The most significant byte (Bits[31:24]) in the 32-bit word is transmitted/received first. This is followed by the next byte (Bits[23:16]) and so on.

The 16-bit data can be packed on 16-bit boundaries. The most significant halfword (Bits[31:16]) is transmitted/received first, followed by the next halfword.

For 12 BPP data on an 8-bit interface, two complete pixels can be accommodated in three bytes. Therefore, in a 32-bit word, the first two colors of the third pixel must be packed in the last byte. The next FIFO word starts with the third color (B3[3:0]) of the third pixel, and so on. Data must be packed as shown in Table 451 and Table 452.

Table 451. 12 BPP Packing for an 8-Bit PDI Interface, Bits[B31:B16]

B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
R1[3:0]				G1[3:0]				B1[3:0]				R2[3:0]			

Table 452. 12 BPP Packing for an 8-Bit PDI Interface, Bits[B15:B0]

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
G2[3:0]				B2[3:0]				R3[3:0]				G3[3:0]			

For 12 BPP data on a 16-bit interface, two complete pixels can be accommodated in a 32-bit word. Data must be packed as shown in Table 453 and Table 454.

Table 453. 12 BPP Packing for a 16-Bit PDI Interface, [B31:B16]

B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
X	X	X	X	R1[3:0]				G1[3:0]				B1[3:0]			

Table 454. 12 BPP Packing for a 16-Bit PDI Interface, [B15:B0]

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	R2[3:0]				G2[3:0]				B2[3:0]			

- Irrespective of the color depth (8/12/16 BPP format), the data transmitted or received over PDI always depends on the data bus width of the PDI. In other words, an 8-bit interface always transmits/receives the first byte (Bits[31:24]) of the PDI_FIFO followed by the next byte (Bits[23:16]) and so on. Similarly, a 16-bit interface always transmits/receives the first halfword (Bits[31:16]) of the PDI_FIFO followed by the next halfword (Bits[15:0]). Therefore, the software takes care of packing the data in the appropriate manner for a specific BPP format.
- For 18 BPP data, there is only one pixel for every word. Each color element (6 bits) of the 18-bit data has to be packed on byte boundaries. This is shown in Table 455 and Table 456.

Table 455. 18 BPP Packing, [B31:B16]

B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
X	X	X	X	X	X	X	X	R1[5:0]						X	X

Table 456. 18 BPP Packing, [B15:B0]

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
G1[5:0]						X	X	B1[5:0]						X	X

Note that in this packing style, each color element is on a byte boundary. This is the best option because it can easily be used with an 8-/16-bit data bus as well for future use.

BLOCK DIAGRAM

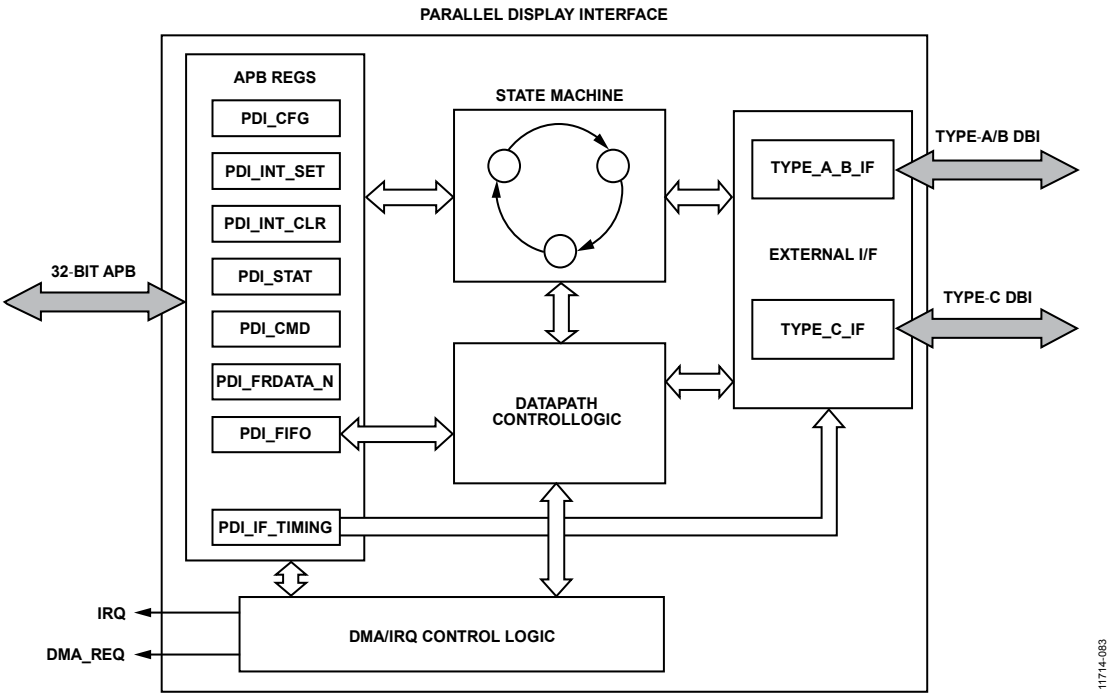


Figure 84. Block Diagram of the PDI Module

PIN LIST

Table 457. Bump Details

Symbol	CSPBGA Bump Location	Name	Direction	Description
RESX	A2	Reset	O	At power-up, the core (via software) initializes the external LCD module by driving the RESX signal.
CSX	B3	Chip select	O	The display module is selected when this signal is low. When CSX is high, the display module ignores all other interface signals.
DCX	A3	Data/command	O	Data is indicated when high and command is indicated when low.
RDX or RWX	B4	Read/write for Type A, read for Type B	O	Type A: Host processor reads data (D[15:0], D[8:0], or D[7:0]) when high or writes data (D[15:0], D[8:0], or D[7:0]) when low. Type-B: Host processor reads information (D[15:0], D[8:0], or D[7:0]) at rising edge.
ECLOCK or WRX	A4	E for Type A, write for Type B	O	Type A: In Fixed E mode, this signal is tied high. In Clocked E mode, the host processor reads information (D[15:0], D[8:0], D[7:0]) at rising edge or writes at falling edge. Type B: Host processor writes information (D[15:0], D[8:0], or D[7:0]) at falling edge.
D[15:0]	A5, B5, A6, B6, A7, B7, A8, B8, A9, B9, A10, B10, A11, B11, A12, B12, D15	Data	I/O	Data lines. If an 8-bit interface is selected, only D[7:0] are valid. Similarly, for a 9-bit interface, only D[8:0] are valid. All lines are valid for a 16-bit interface.
TE	D15	Tearing effect	I	Tearing effect signal.
SCL	A5	Serial clock	O	Host processor writes information (DOUT) at falling edge or reads information (DIN) at rising edge.
DOUT	B5	Serial data out	O	Serial data output from the PDI.
DIN	A6	Serial data in	I	Serial data input to the PDI.

Note that the VLCD pin must be connected to DVDD when in PDI mode.

FUNCTIONAL OPERATION

The PDI module is programmable via the 32-bit APB interface. It has a set of registers for this purpose that can be used to enable/disable different features of the module. It has a DMA_EN bit in the PDI_CFG register to configure the DMA request generation. It also has registers to program the timing characteristics of the assertion/deassertion of various interface signals to support a wide variety of LCD modules.

Initialization Sequence

The LCD module needs to be initialized before starting any data transactions. At power-up, the core (via software) initializes the external LCD module by driving the RESX signal. Because the PDI does not control the RESX pin, this signal is mapped to a GPIO pin. Software must configure the assigned GPIO as an output and drive it appropriately. The GPT on the ADuCM350 can be used to control the timing related to the reset signal.

PROGRAMMING DBI COMMANDS

PDI_EN Bit

This bit in the PDI_CFG register is the master enable for the PDI device. All clocks used internally are gated by this bit. This bit must be set before an APB register access can occur. Clearing this bit acts like a software reset for the block. Internally, the module clears all state machines after cleanly exiting the ongoing DBI transfers (if any).

PDI Interface Timing Dependencies

The user must program the timing parameters in such a way that the parameters adhere to the MIPI DBI ac characteristics. All timing options are available in integer multiples of the HCLK period. However, the value of T mentioned in the MIPI DBI specification needs not be the same as the HCLK period. Therefore, it is the user's responsibility to choose the appropriate value for each PDI_IF_TIMING field.

Moreover, the values programmed must be consistent. For example, from the MIPI specification, it is evident that the sum of the address setup and address hold times must not be greater than TWRH_RDH. That is,

$$2 \times TAS\_AH \leq TWRH\_RDH$$

Care must be taken for the TAH timing for the current cycle and the TAS timing for the next cycle. Therefore, if the user wants a higher setup/hold time, program a bigger TWRH_RDH time as well.

DBI Programming Sequence

The core can write the command and other related fields (such as R/W, type of parameter, and the number of parameters following the command) to the PDI_CMD register. Parameters are always written to or read from the PDI_FIFO register. All FIFO accesses must be 32 bits wide. Lesser width accesses, like 16-bit or 8-bit accesses, are not supported.

For write commands other than a frame data write, the parameters (if any) have to be written by the core to the PDI_FIFO register only after writing into the PDI_CMD register. This is necessary because the direction of the FIFO is determined by the WR_RD bit in the PDI_CMD register. It is also necessary to clear the FR_DATA bit in the PDI_CMD register for these commands. As soon as there is valid data in the PDI_FIFO register, the controller starts driving the command followed by the parameter over DBI. All control parameters (parameters that are not frame data) are transmitted as bytes, irrespective of the PDI data-width (8-/9-/16-bit). That is, only the lower 8 bits of the data bus are used for transmit per the MIPI DCS specification.

For read commands other than frame data read, the received parameters are written to the PDI_FIFO register by the PDI controller. The FR_DATA bit must be cleared while writing to the PDI_CMD register. The data received can be read by the core while servicing the interrupt (RX_IRQ or DONE_IRQ) or after polling for FIFO status. However, if DMA service is not required for this command, the core must clear the DMA_EN bit before writing to the PDI_CMD register. All control parameters are received as bytes, regardless of the PDI data width (8-/9-/16-bit). That is, only the lower eight bits of the data bus are used for receive per the MIPI DCS specification.

All frame data write/read commands (such as READ_MEMORY_START, READ_MEMORY_CONTINUE, WRITE_MEMORY_START, and WRITE_MEMORY_CONTINUE) can be programmed by the core after enabling the DMA requests. It is also necessary to set the FR_DATA bit in the PDI_CMD register for these commands. All data requests can be serviced by the DMA controller by writing to or by reading from PDI_FIFO. The data width of each DMA transfer is 32 bits. By default, all four bytes in a single FIFO location are decoded depending on the DBI bus width and BPP format. If the frame data chunk is not aligned to a word boundary (that is, a transfer must end with 1, 2, or 3 bytes of the last 32-bit word), the user must program the number of parameters in the PDI_FRDATA_N register (13 bits wide). If the frame data chunk is aligned to a word boundary, this step is not necessary. After the transaction has ended, the PDI_FIFO is flushed to clear any unused data.

For frame data writes, the PDI controller waits for new valid data in the PDI_FIFO register before initiating the command. After starting the data transfer, if the FIFO is empty, this module pauses the write command (by deasserting CSX). It continues with the writes only when there is valid data in the Tx FIFO.

For frame data reads, the PDI controller updates the PDI_FIFO register with the data received over the DBI. The DMA must read this data to clear up FIFO space. If the FIFO is full, the controller pauses the read command (by deasserting CSX). It continues with the reads when there is space in the FIFO.

Note the following:

- The direction of the DBI command is obtained from the WR_RD bit of the PDI_CMD register. This determines if the PDI_FIFO is going to be a Tx/Rx FIFO.
- The PDI controller is not restricted to generate DMA requests only for frame data. This is because there are some other commands (such as WRITE_LUT, READ_DDB_START, and READ_DDB_CONTINUE) that may have a large number of parameters. These commands may be supported in the future and they may use DMA.
- A single DMA transfer is limited to 1024 words. Therefore, for a single frame data write/read, the user may have to perform multiple DMA cycles. However, the user may choose to program the value of the FR_DATA_N bit for the last DMA cycle only. In this case, the user does not have to write to the PDI_CMD register for each DMA cycle.

PSEUDOCODE EXAMPLE

DMA Transmit Configuration

The DMA PDI channel must be configured for a transmit operation. The NVIC must be configured to enable DMA PDI interrupt.

For example, consider Type B DBI with an 8-bit D bus and 8 BPP color depth. The PDI must be configured as follows:

- PDI_CFG = 0x00000105; // Enables PDI in Type B mode. DMA is enabled.
- PDI_IF_TIMING = 0x01010111; // Configures ac timing values for the DBI interface.
- PDI_FRDATA_N = 0x000003FF; // Configures the number of bytes to be transmitted over DBI. In this example, the number of bytes is 1023.
- PDI_CMD = 0x0000032C; // Programs a WRITE_MEMORY_START command (2C) and write direction.

The PDI initiates DMA requests whenever there is space in the Tx FIFO. After all the transfers are complete, the DONE interrupt is triggered by the DMA controller. The next set of data can then be programmed.

DMA Receive Configuration

The DMA PDI channel must be configured for a receive operation. The NVIC must be configured to enable DMA PDI interrupt.

For example, consider Type C DBI with an 8-bit serial data. The PDI must be configured as follows.

- PDI_CFG = 0x00000109; // Enables PDI in Type C mode and DMA is enabled.
- PDI_IF_TIMING = 0x01010111; // Configures ac timing values for the DBI interface.
- PDI_FRDATA_N = 0x000001FF; // Configures the number of bytes to be Rxed over DBI. Here, it is 511.
- PDI_CMD = 0x0000022E; // Programs a READ_MEMORY_START command (2E) and the direction as read.

The PDI initiates DMA requests whenever there is data available in the Rx FIFO. After all the transfers are done, the DONE interrupt is triggered by the DMA controller. The next set of data can then be programmed.

Core Transmit Configuration

The NVIC must be configured to enable PDI interrupt.

For example, consider Type A (Fixed E mode) DBI with a 9-bit D bus and 18 BPP color depth. The PDI must be configured as follows.

- PDI_CFG = 0x00000071; // Enables PDI in Type A fixed E mode, 9-bit D bus and 18 BPP. DMA is disabled.
- PDI_IF_TIMING = 0x01010111; // Configures ac timing values for the DBI interface.
- PDI_INT_SET = 0x00000008; // Enables DONE interrupt to indicate the end of transfers.
- PDI_CMD = 0x00020136; // Programs the DBI command and the direction as write. In this example, the number of parameters is 2.
- PDI_FIFO = 0xABAB0000; // Writes the two parameters that must be transmitted after the DBI command.

The PDI transmits the DBI command followed by the two parameters. After this is done, the IRQ line is asserted by the PDI module. In the ISR, the core can read the PDI_STAT register to determine the cause of the interrupt. In this case, it is DONE_IRQ. To clear this interrupt, the core must write a 1 to this bit in the PDI_STAT register. Alternatively, the user can check whether transfers are complete by polling the FIFO status.

Core Receive Configuration

The NVIC must be configured to enable a PDI interrupt.

For example, consider Type A (Fixed E mode) DBI with a 9-bit D bus and 18 BPP color depth. The PDI must be configured as follows.

- PDI_CFG = 0x00000071; // Enables PDI in Type A Fixed E mode, 9-bit D bus and 18 BPP. DMA is disabled.
- PDI_IF_TIMING = 0x01010111; // Configures ac timing values for the DBI interface.
- PDI_INT_SET = 0x00000002; // Enables Rx interrupt.
- PDI_CMD = 0x0001000E; // Programs the DBI command and the direction as read. In this example, the number of parameters is 1.

The PDI transmits the DBI command. It then receives the parameter associated with the command. When this data is written to the FIFO, the Rx interrupt is triggered by the PDI module. The core can then read the PDI_STAT register to determine the cause of the interrupt, and then it can read the PDI_FIFO register. This interrupt is cleared automatically when the FIFO becomes empty.

DMA BANDWIDTH ANALYSIS FOR VIDEO DATA

Table 458 shows a comparison of the required DMA bandwidth for the different LCD screen resolutions and color depths. Because all of the DMA transfers are 32 bits wide, each DMA transfer/sec has 4 pixels/word for an 8 BPP format and 2 pixels/word for a 12 BPP (and 16 BPP) format.

Table 458. DMA Bandwidth Required for 15 fps Video

			DMA Transfers/Sec	
Screen Number	**Resolution**	**Total Number of Pixels/Frame**	**8-Bit Color (4 Pixels/Word)**	**12-Bit Color (2 Pixels/Word)**
1	320 × 480	153,600	576k	1.15M
2	240 × 430	103,200	387k	774k
3	240 × 320	76,800	288k	576k

PDI MEMORY MAPPED REGISTERS

PDI Register Map

Table 459. PDI Register Summary

Address	Name	Description	Reset	RW
0x40030000	PDI_CFG	PDI configuration register	0x00000000	RW
0x40030004	PDI_INT_SET	PDI interrupt set register	0x00000000	W
0x40030008	PDI_INT_CLR	PDI interrupt clear register	0x00000000	W
0x4003000C	PDI_STAT	PDI status register	0x00000040	RW1C
0x40030010	PDI_CMD	PDI command register	0x00000000	RW
0x40030014	PDI_FRDATA_N	PDI frame data count register	0x00000000	RW
0x40030018	PDI_FIFO	PDI parameter FIFO	0x00000000	RW
0x4003001C	PDI_IF_TIMING	PDI interface timing register	0x01010111	RW

PDI Configuration Register

Address: 0x40030000, Reset: 0x00000000, Name: PDI_CFG

Table 460. Bit Descriptions for PDI_CFG

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0000	RESERVED
[14:10]	RESERVED	Reserved.	0x00	RESERVED
9	END_XFER	End DBI transfer. This bit can be used to end a frame data transfer when FR_DATA_N = 0. If this bit is set while a DBI Tx is going on, PDI ends the DBI transfer after the FIFO becomes empty. It does not wait for any more data. If this is set during a DBI Rx, PDI ends the transfer immediately (irrespective of the FIFO status). Note that if FR_DATA_N is not 0, this bit must be used only when a DBI write/read must be ended midway. This is more of a recovery feature for a clean exit. For a clean shut off of PDI, the s/w must set this bit, clear the DMA_EN bit, and then clear the PDI_EN bit (in the same order). 0: DBI transfer continues. 1: DBI transfer is ended.	0x0	RWAC
8	DMA_EN	DMA request enable. 0: DMA requesting not enabled. 1: DMA requesting enabled.	0x0	RW
7	FLSH_FIFO	Flush the parameter FIFO. 0: DBI parameter FIFO is not flushed. 1: DBI parameter FIFO is flushed.	0x0	RWAC

Bits	Bit Name	Description	Reset	Access
[6:4]	DWIDTH_BPP	Data bus width and color depth. This field is invalid if Type C (serial) interface is selected in the DBI_TYPE field. 000: 8-bit D bus and 8 BPP color depth. 001: 8-bit D bus and 12 BPP color depth. 010: 8-bit D bus and 16 BPP color depth. 011: 16-bit D bus and 8 BPP color depth. 100: 16-bit D bus and 12 BPP color depth. 101: 16-bit D bus and 16 BPP color depth. 11x: 9-bit D bus and 18 BPP color depth.	0x0	RW
[3:1]	DBI_TYPE	Type of the DBI. 000: Type A and Fixed E mode. 001: Type A and Clocked E mode. 010: Type B. 011: Type C Option 1 (9-bit). 1XX: Type C Option 3 (8-bit).	0x0	RW
0	PDI_EN	PDI controller enable. 0: disable PDI controller. 1: enable PDI controller.	0x0	RW

PDI Interrupt Set Register

Address: 0x40030004, Reset: 0x00000000, Name: PDI_INT_SET

Read of this register gives the actual status of the interrupt enable bits.

Table 461. Bit Descriptions for PDI_INT_SET

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	RESERVED
11	SET_FIFO_UDF_IEN	Enable FIFO underflow IRQ. 0: FIFO underflow IRQ enable is not changed. 1: FIFO underflow IRQ is enabled.	0x0	W
10	SET_FIFO_OVF_IEN	Enable FIFO overflow IRQ. 0: FIFO overflow IRQ enable is not changed. 1: FIFO overflow IRQ is enabled.	0x0	W
9	SET_CMD_WR_ERR_IEN	Enable command write error IRQ. 0: command write error IRQ enable is not changed. 1: command write error IRQ is enabled.	0x0	W
8	SET_CMD_DONE_IEN	Enable command done IRQ. 0: command done IRQ enable is not changed. 1: command done IRQ is enabled.	0x0	W
[7:4]	RESERVED	Reserved.	0x0	RESERVED
3	SET_DONE_IEN	Enable done IRQ. If enabled, the controller generates an interrupt after all the N parameters have been transmitted/received over DBI. 0: done IRQ enable is not changed. 1: done IRQ is enabled.	0x0	W
2	SET_TE_IEN	Enable tearing effect (TE) IRQ. This enables the generation of an interrupt whenever the TE signal is asserted. Note that no differentiation is made between VSYNC and HSYNC pulses. 0: TE IRQ enable is not changed. 1: TE IRQ is enabled.	0x0	W
1	SET_RX_IEN	Enable Rx IRQ. 0: Rx IRQ enable is not changed. 1: Rx IRQ is enabled.	0x0	W
0	SET_TX_IEN	Enable Tx IRQ. 0: Tx IRQ enable is not changed. 1: Tx IRQ is enabled.	0x0	W

PDI Interrupt Clear Register

Address: 0x40030008, Reset: 0x00000000, Name: PDI_INT_CLR

Read of this register gives the actual status of the interrupt enable bits.

Table 462. Bit Descriptions for PDI_INT_CLR

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	RESERVED
11	CLR_FIFO_UDF_IEN	Disable FIFO underflow IRQ. 0: FIFO underflow IRQ enable is not changed. 1: FIFO underflow IRQ is disabled.	0x0	W
10	CLR_FIFO_OVF_IEN	Disable FIFO overflow IRQ. 0: FIFO overflow IRQ enable is not changed. 1: FIFO overflow IRQ is disabled.	0x0	W
9	CLR_CMD_WR_ERR_IEN	Disable command write error IRQ. 0: command write error IRQ enable is not changed. 1: command write error IRQ is disabled.	0x0	W
8	CLR_CMD_DONE_IEN	Disable command done IRQ. 0: command done IRQ enable is not changed. 1: command done IRQ is disabled.	0x0	W
[7:4]	RESERVED	Reserved.	0x0	RESERVED
3	CLR_DONE_IEN	Disable done IRQ. 0: done IRQ enable is not changed. 1: done IRQ is disabled.	0x0	W
2	CLR_TE_IEN	Disable TE IRQ. 0: TE IRQ enable is not changed. 1: TE IRQ is disabled.	0x0	W
1	CLR_RX_IEN	Disable Rx IRQ. 0: Rx IRQ enable is not changed. 1: Rx IRQ is disabled.	0x0	W
0	CLR_TX_IEN	Disable Tx IRQ. 0: Tx IRQ enable is not changed. 1: Tx IRQ is disabled.	0x0	W

PDI Status Register

Address: 0x4003000C, Reset: 0x00000040, Name: PDI_STAT

This register gives the status of interrupts to the core and the FIFO status.

Table 463. Bit Descriptions for PDI_STAT

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	RESERVED
11	FIFO_UDF	FIFO underflow. This bit indicates that a FIFO read was attempted when it is empty. Both read and write pointers remain unchanged. This bit is cleared only when 1 is written to this bit. If FIFO_UDF IRQ is enabled, this bit generates an interrupt. 0: FIFO underflow IRQ did not occur. 1: FIFO underflow IRQ occurred.	0x0	RW1C
10	FIFO_OVF	FIFO overflow. This bit indicates that a FIFO write was attempted when it is full. When the FIFO is full, any more data written is ignored. Both read and write pointers remain unchanged. This bit is cleared only when 1 is written to this bit. If FIFO_OVF IRQ is enabled, this bit generates an interrupt. 0: FIFO overflow did not occur. 1: FIFO overflow occurred.	0x0	RW1C

Bits	Bit Name	Description	Reset	Access
9	CMD_WR_ERR	PDI command write error. This bit indicates that the PDI_CMD register was written while there is an ongoing PDI transaction. This bit is cleared only when 1 is written to this bit. If CMD_WR_ERR IRQ is enabled, this bit generates an interrupt. 0: command write error did not occur. 1: command write error occurred.	0x0	RW1C
8	CMD_DONE	PDI command completed. This bit gets asserted whenever a PDI command transmission completes. When programming back to back PDI commands which have no parameters associated, this bit must be checked after each PDI_CMD write. This is to ensure that the succeeding PDI command write doesn't corrupt the ongoing transfer. This bit is cleared only when 1 is written to this bit. If CMD_DONE IRQ is enabled, this bit generates an interrupt. Note that if the current PDI command has parameters associated with it, the user must check for the done IRQ before programming the next PDI command. 0: PDI command is not completed. 1: PDI command is completed.	0x0	RW1C
7	FIFO_FULL	FIFO full. 0: FIFO is not full. 1: FIFO is full.	0x0	R
6	FIFO_EMPTY	FIFO empty. 0: FIFO is not empty. 1: FIFO is empty.	0x1	R
[5:4]	FIFO_STAT	FIFO status. 00: If FIFO_EMPTY is 1, there are no words in FIFO. Otherwise, there is one word in FIFO. 01: two words in FIFO. 10: three words in FIFO. 11: four words in FIFO.	0x0	R
3	DONE_IRQ	Done interrupt. If DONE_IRQ is enabled, this interrupt occurs after a DBI transfer is completed; that is, after transmitting/receiving the DBI command and all the parameters (control parameters or frame data) associated with it. For frame data transfers where FR_DATA_N = 0, the completion of DBI transfer can happen when the PDI_EN is cleared or when END_XFER is set. In both cases, this bit is set. This interrupt is cleared only when 1 is written to this bit. Note that for commands that have no associated parameters, this bit/IRQ is not set. 0: DONE_IRQ did not occur. 1: DONE_IRQ occurred.	0x0	RW1C
2	TE_IRQ	Tearing effect interrupt. If TE_IRQ is enabled, this interrupt occurs whenever a rising edge is detected in the TE signal. This interrupt is cleared only when 1 is written to this bit. 0: TE_IRQ did not occur. 1: TE_IRQ occurred.	0x0	RW1C
1	RX_IRQ	Rx interrupt. If RX_IRQ is enabled, this interrupt occurs whenever there is a data available in the FIFO (FIFO not empty) during a read transfer. It is cleared when the FIFO becomes empty or if CLR_RX_IEN is set. 0: RX_IRQ did not occur. 1: RX_IRQ occurred.	0x0	R
0	TX_IRQ	Tx interrupt. If TX_IRQ is enabled, this interrupt occurs whenever there is a space available in the FIFO (FIFO not full) during a write transfer. It is cleared when the FIFO becomes full or if CLR_TX_IEN is set. 0: TX_IRQ did not occur. 1: TX_IRQ occurred.	0x0	R

PDI Command Register

Address: 0x40030010, Reset: 0x00000000, Name: PDI_CMD

Table 464. Bit Descriptions for PDI_CMD

Bits	Bit Name	Description	Reset	Access
[31:24]	RESERVED	Reserved.	0x00	RESERVED
[23:16]	N_PARAM	Number of control parameters. Specifies the number of control parameter bytes that are associated with a DBI command.	0x00	RW
[15:10]	RESERVED	Reserved.	0x00	RESERVED
9	FR_DATA	Type of parameter. 0: control parameter. 1: frame data.	0x0	RW
8	WR_RD	Write or read direction. 0: read command. 1: write command.	0x0	RW
[7:0]	CMD	DBI command. This is the 8-bit DBI command. A write to this register triggers the DBI transfer.	0x00	RW

PDI Frame Data Count Register

Address: 0x40030014, Reset: 0x00000000, Name: PDI_FRDATA_N

Note that this register is internally cleared as soon as the DBI transfer ends.

Table 465. Bit Descriptions for PDI_FRDATA_N

Bits	Bit Name	Description	Reset	Access
[31:13]	RESERVED	Reserved.	0x00000	RESERVED
[12:0]	FR_DATA_N	Number of DBI frame data transfers. This register gives the number of bytes for an 8-bit Type A/Type B interface or for a Type C interface. For a 9-/16-bit Type A/Type B interface, this refers to the number of 9-/16-bit transfers, respectively. By default, when this register is 0, the DBI transfers (Tx/Rx) are continuously generated, unless the END_XFER bit is set. If a nonzero value is written, then DBI transfers are generated only until FR_DATA_N DBI words are transmitted/received. Note that because the DMA can do up to 1024 words, this number can be up to 4096 bytes. So, all values greater than 13'h1000 are illegal.	0x0000	RW

PDI Parameter FIFO Register

Address: 0x40030018, Reset: 0x00000000, Name: PDI_FIFO

Table 466. Bit Descriptions for PDI_FIFO

Bits	Bit Name	Description	Reset	Access
[31:0]	PARAM	Parameters that follow DBI command. This is a 4-deep FIFO (32-bits each location) which are used to transmit/receive parameters of all kind (control parameters and frame data). The FIFO is flushed automatically at the end of a DBI transfer. Note that all accesses to this register must always be no less than 32-bits wide.	0x00000000	RW

PDI Interface Timing Register

Address: 0x4003001C, Reset: 0x01010111, Name: PDI_IF_TIMING

Table 467. Bit Descriptions for PDI_IF_TIMING

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved.	0x0	RESERVED
[27:24]	TCSH	Chip select setup time. Number of HCLK cycles required for chip select setup time for read in Type C interface. It can range from 1 to 15 HCLK cycles. Programming a value of 0 is illegal.	0x1	RW
[23:16]	TWRL_RDL	Low duration for write/read control pulse. Number of HCLK cycles required for the low duration of the write/read control pulse. Both the write and read operations have the same value. For Type A, this value corresponds to the first half of the t_{CYCLE} period. It can range from 1 to 255 HCLK cycles. Programming a value of 0 is illegal.	0x01	RW
[15:8]	TWRH_RDH	High duration for write/read control pulse. Number of HCLK cycles required for the high duration of the write/read control pulse. Both the write and read operations have the same value. For Type A, this value corresponds to the second half of the t_{CYCLE} period. It can range from 1 to 255 HCLK cycles. Programming a value of 0 is illegal.	0x01	RW
[7:4]	TCSF	Chip select wait time. Number of HCLK cycles required for chip select wait time in Type B interface. Default value of 1 HCLK cycle is used. It can range from 1 to 15. Programming a value of 0 is illegal. Note that as per DBI specification, this value is in the order of 20 ns and, thus, it is not expected to be changed from 1.	0x1	RW
[3:0]	TAS_AH	Address setup/hold time. Number of HCLK cycles required for address setup/hold phases. Both the setup and hold times have the same value. It can range from 1 to 15 cycles. Programming a value of 0 is illegal.	0x1	RW

AFE EXCITATION LOOP

INTRODUCTION

In this section, an overview of the analog front-end excitation loop is outlined. See Figure 86 and Figure 87 for block diagrams of the loop. The AFE excitation loop consists of an excitation buffer amplifier loop, transimpedance amplifier (TIA), and switch matrix. The switch matrix is a programmable interface with an external sensor configuration and a calibration resistor. Typically, a calibration resistor has two functions:

- Allow ratiometric measurements during ac excitation for calibration. This allows for gain and offset errors in the loop to be calibrated out.
- Allows generation of an accurate current source for calibration of ADC channel preceding dc excitation. See the No Factory Calibration section for more information.

The AFE excitation loop sets the voltage across a sensor to be equal to twice the differential DAC output voltage. The gain of 2 is achieved by making the R1B/R1A ratio = 2. As the force and sense points differ, it provides an accurate method of controlling the voltage across the sensor through access resistances. The voltage across the sensor sets up an impedance dependent current to flow to the transimpedance amplifier. The transimpedance amplifier converts the current signal to a voltage using an external resistor, and this voltage can then be measured by the ADC, giving the ability to measure the impedance.

A brief description of the primary functions of each block within the AFE excitation loop follows:

- Excitation amplifier loop.
  - Sources necessary sensor excitation current by buffering the filtered DAC voltage.
  - Provides gain to AFE excitation loop to accurately set the sensor excitation voltage.
  - The feedback capacitors and resistors in conjunction with the gain from the excitation buffer place a zero in the loop to increase the gain excitation frequencies.
  - High impedance inputs to give accurate sensing of excitation voltage through sense access resistance of the sensor.
- Transimpedance amplifier.
  - Current to voltage conversion for measurement by ADC, the gain of which is set by an external resistor.
  - Sinks sensor excitation current.
  - Accurate setting of the working electrode voltage, which is the common mode for the sensor.
  - Antialiasing filtering preceding the ADC.
- Switch matrix.
  - Allows calibration of an unknown sensor impedance vs. an external known resistor.
  - Allows measurement of sensor access resistances, switch resistances, and leakages for calibration sensor parameter limits.

SWITCH MATRIX

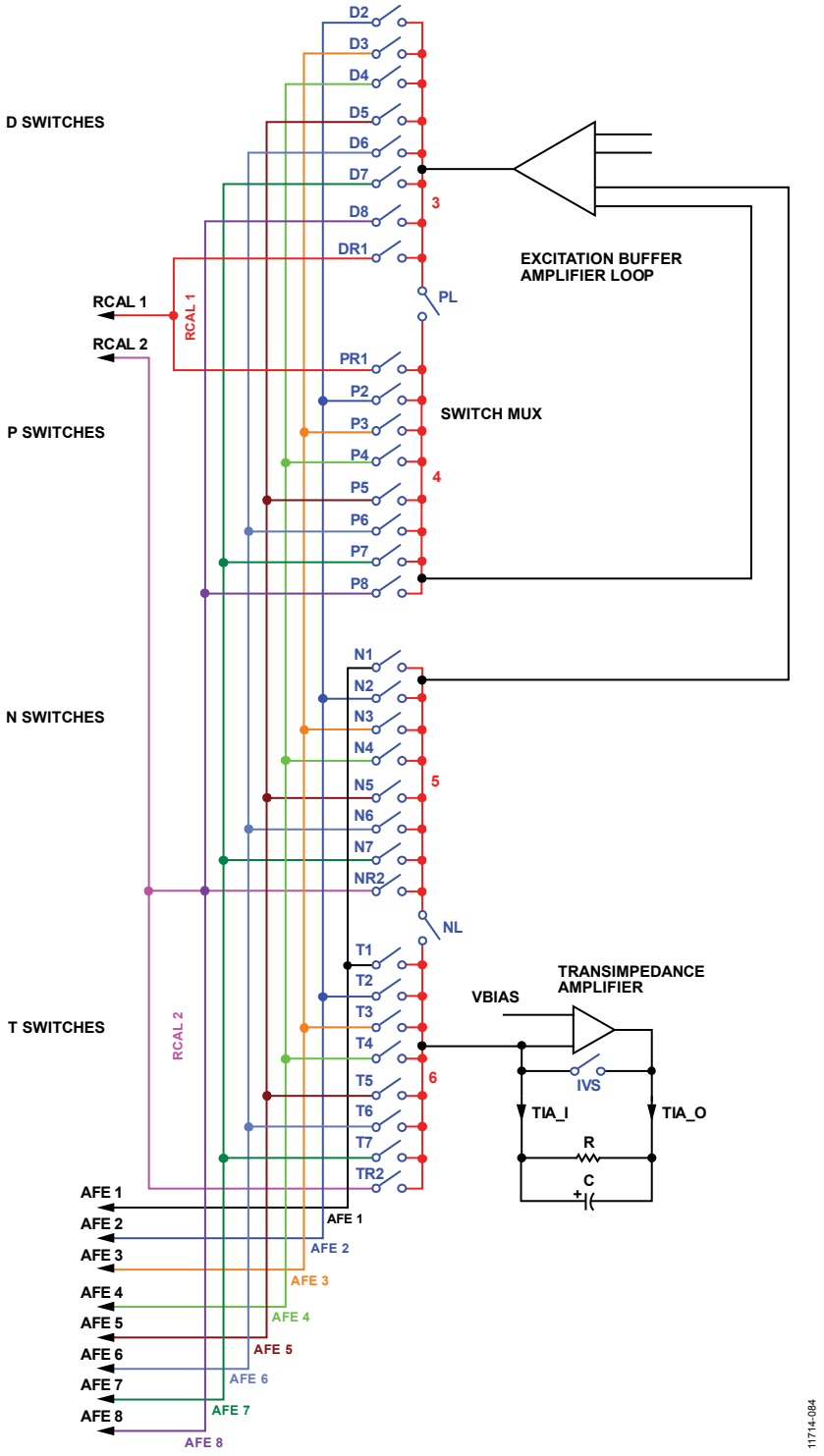


Figure 85. Switch Matrix Configuration

Individual control of each switch in the switch matrix is possible by programming the bits in the AFE_SW_CFG register, as described in the Analog Front-End Interface section.

BASIC LOOP THEORY

In Figure 86, a sensor is hooked up in a 2-wire sensor configuration. For 2-wire or 3-wire sensor configurations, the P and N sense lines can be tied internally through the switch matrix.

The signal is switched through AFE 8 (Switch D8 closed; see Figure 85 for details of the switch matrix) to the counter electrode of the sensor. The working electrode is held at a fixed common mode at AFE 1 (Switch T1 closed). This results in a current flow from the D node through the sensor to the T node. This current is converted to a voltage using the R_{TIA} on the feedback path of the transimpedance amplifier (TIA).

$$Current\ into\ T\ Node = (V_{AFE\ 8} - V_{AFE\ 1})/(R_A + Z_{SENSOR} + R_A)$$

$$Voltage\ Seen\ at\ ADC = Current\ into\ T\ Node \times R_{TIA}$$

- The stimulus DAC voltage, described in more detail in the DAC chapter, has an output voltage of V_{DAC} from 0.2 V to 1.0 V.
- The DAC output voltage is reference to 0.6 V by the in-amp reference.
- This signal is then gained by two in the excitation loop stage, Amplifier A1, in Figure 86.
- The common mode of the system is set up by the positive input terminal of the TIA, which is tied to VBIAS of 1.1 V by default.

The governing equation for the voltage appearing at the output, AFE 8, is $V_{AFE\ 8} = ((V_{DAC} - 0.6\ V) \times 2) + VBIAS$.

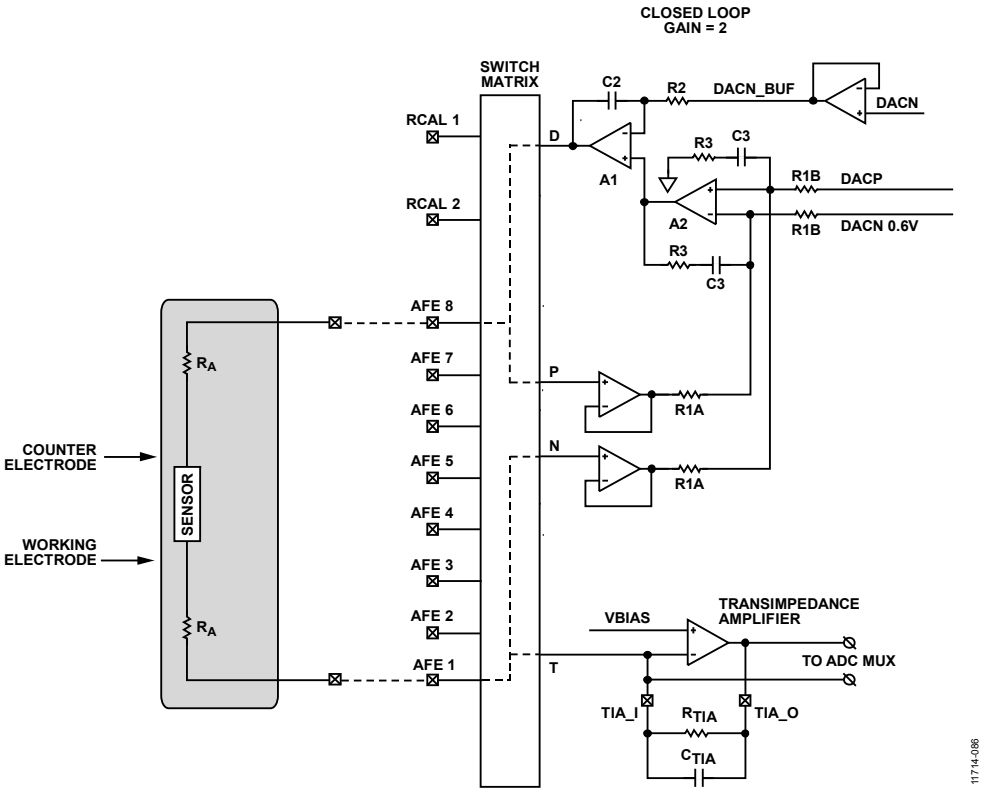


Figure 86. AFE Excitation Loop Block Diagram, 2-Wire Configuration

Consider a 4-wire configuration, as shown in Figure 87. The sensor is being sensed differentially by the P and N nodes. Both P and N are fed back into the feedback loop of the excitation stage through an instrumentation amplifier using AFE 6 (Switch P6 closed) and AFE 3 (Switch N3 closed). The excitation buffer forces the output voltage on D to be equal to the differential voltage across the sensor, P – N. Both P and N switches are very high input impedance, which guarantees no current flows in sense lines. The common mode of the system is still set by the positive terminal of the TIA, VBIAS of 1.1 V by default.

This single stimulus/differential sense configuration allows a very accurate amperometric configuration system.

Current into T Node = $(V_{COUNTER} - V_{WORKING})/Z_{SENSOR}$

Voltage Seen at ADC = Current into T Node × R_{TIA}

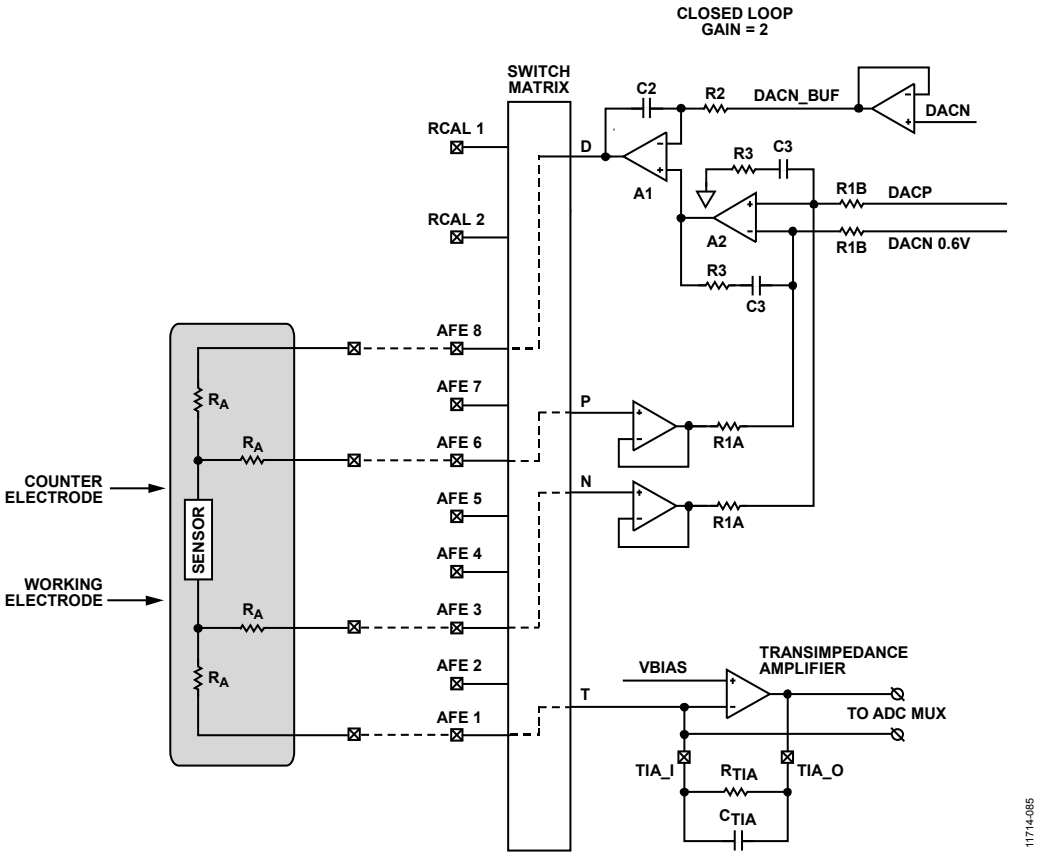


Figure 87. AFE Excitation Loop Block Diagram, 4-Wire Configuration

How to Use the TIA to Set the AFE Loop Common Mode

This explanation uses the 2-wire configuration for simplicity, but the same theory holds for 4-wire configuration. The TIA positive input terminal, 1.1 V by default, sets the common mode of the system. If the positive terminal of TIA is set at 1.1 V, VBIAS, then the negative input terminal is also set to 1.1 V. This means the T node of the switch matrix is set to 1.1 V. T and N are tied internally in the switch matrix so the N node must also be tied to 1.1 V. The excitation buffer forces the D node so that P – N = 0 V, which ensures that the excitation buffer drives the P node to equal the V_{BIAS} of 1.1 V. The D node is tied internally to the P node so the D node is driven to a common mode VBIAS of 1.1 V.

If a step voltage of 300 mV is desired on the sensor, the TIA common mode remains at 1.1 V. The excitation buffer forces the D node so that P – N = 300 mV. The excitation buffer drives the output so that the P sense line sees 1.4 V.

AFE SIGNAL SWINGS

This section illustrates the signal swings that are present on the AFE during measurement sequences

Transmit Stage

The excitation channel consists of a 12-bit string DAC, a reconstruction amplifier (RCF), a programmable gain amplifier (PGA), an excitation loop, a switch matrix, a TIA, and a TIA switch matrix. The PGA has a gain of 1 or 0.025, depending on whether DAC_ATTEN_EN is equal to 0 or 1, respectively. Figure 89 shows example signal swings when DAC_ATTEN_EN = 0. The DAC core has an excess of 200 mV range to allow digital offset and gain correction. The maximum swing that the amplifiers are designed for is 1.2 V p-p at RCAL/sensor. Note how the closed loop gain of the excitation loop is 2. For more information, see the DAC Signal Generation Requirements section.

Nonattenuator Mode

The following characteristics apply to nonattenuator mode:

- 12-bit DAC
- Recommendation to use only 1/8 to 7/8 of DAC range
- Overranged for offset correction
- 1.2 V p-p output range
- LSB size = 1.6 V/2^{12} = 390.63 µV

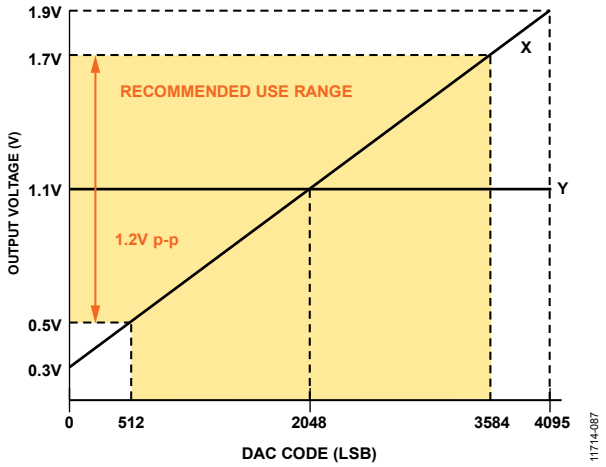


Figure 88. DAC Transfer Function for Nonattenuator Mode

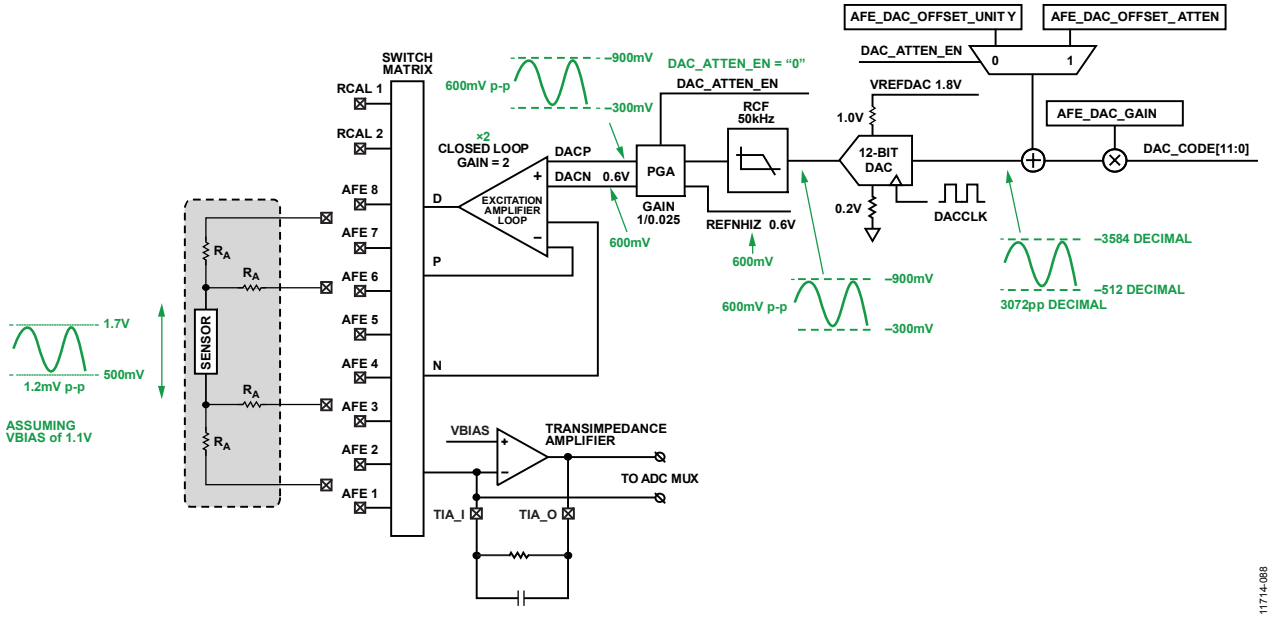


Figure 89. AFE Excitation Channel Showing Example Signal Swings When DAC_ATTEN_EN = 0, Used for DC Excitation

Attenuator Mode

The following characteristics apply to attenuator mode:

- 12-bit DAC
- Recommendation to use only 1/8 to 7/8 of DAC range
- Overranged for offset correction
- 30 mV p-p output range
- LSB size = 1/40 (1.6 V/2^{12})= 9.77 µV

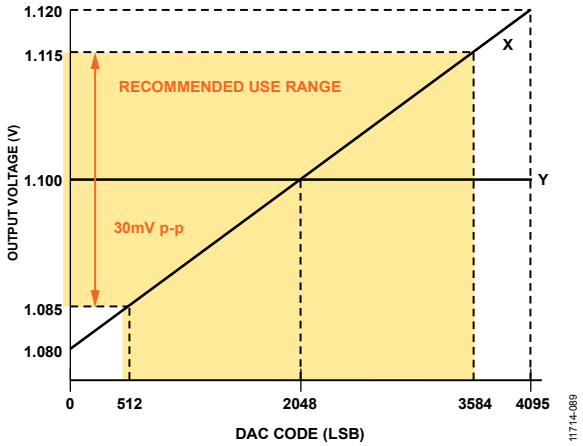


Figure 90. DAC Transfer Function for Attenuator Mode

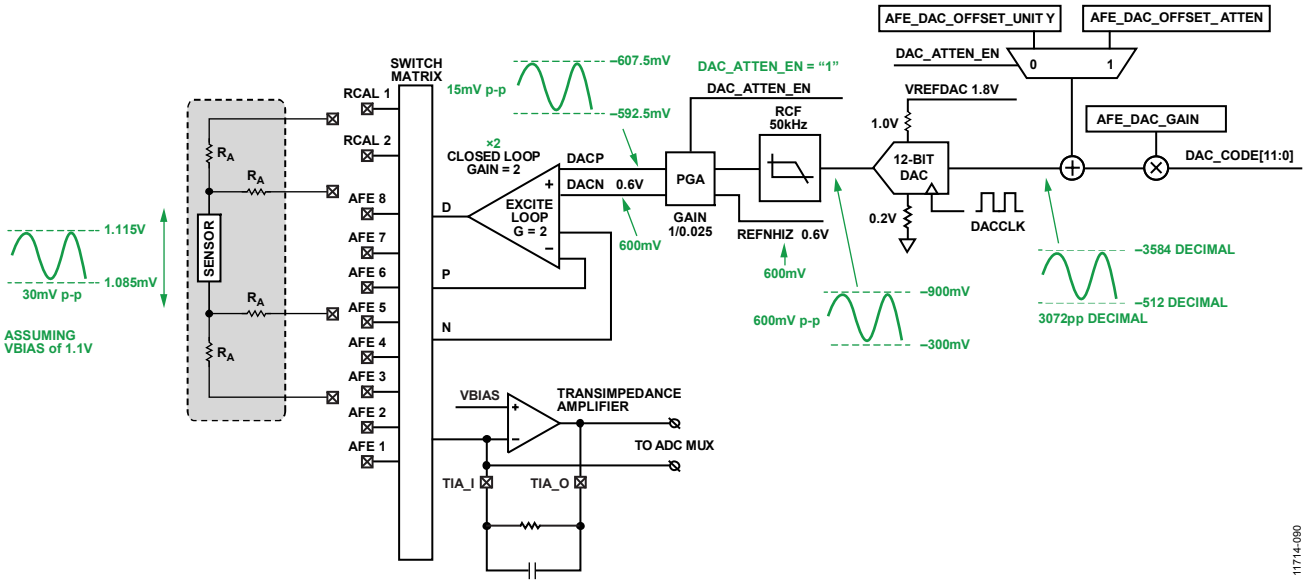


Figure 91. AFE Excitation Channel Showing Example Signal Swings When DAC_ATTEN_EN = 1 Used for AC Excitation

RECEIVE STAGE

For information about the receive stage, see the ADC Channel section.

ANALOG FRONT-END INTERFACE

FEATURES

The AFE interface provides connectivity between the Cortex-M3 system in the ADuCM350 and the measurement AFE. The AFE consists of a collection of analog blocks such as the switch matrix, DAC, ADC, and temperature sensor. It also contains custom DSP hardware accelerators and a programmable sequencer (see Figure 92).

The AFE interface features on the ADuCM350 include the following:

- Full access to analog and DSP blocks (controllably and observable)
- Independent operation not subject to system load
- Offloading of timing critical operations from the Cortex-M3
- Programmable sequencer for cycle accurate AFE functions
- Custom DSP hardware accelerators
- DMA channels for data transfers
- Interrupt from user maskable sources

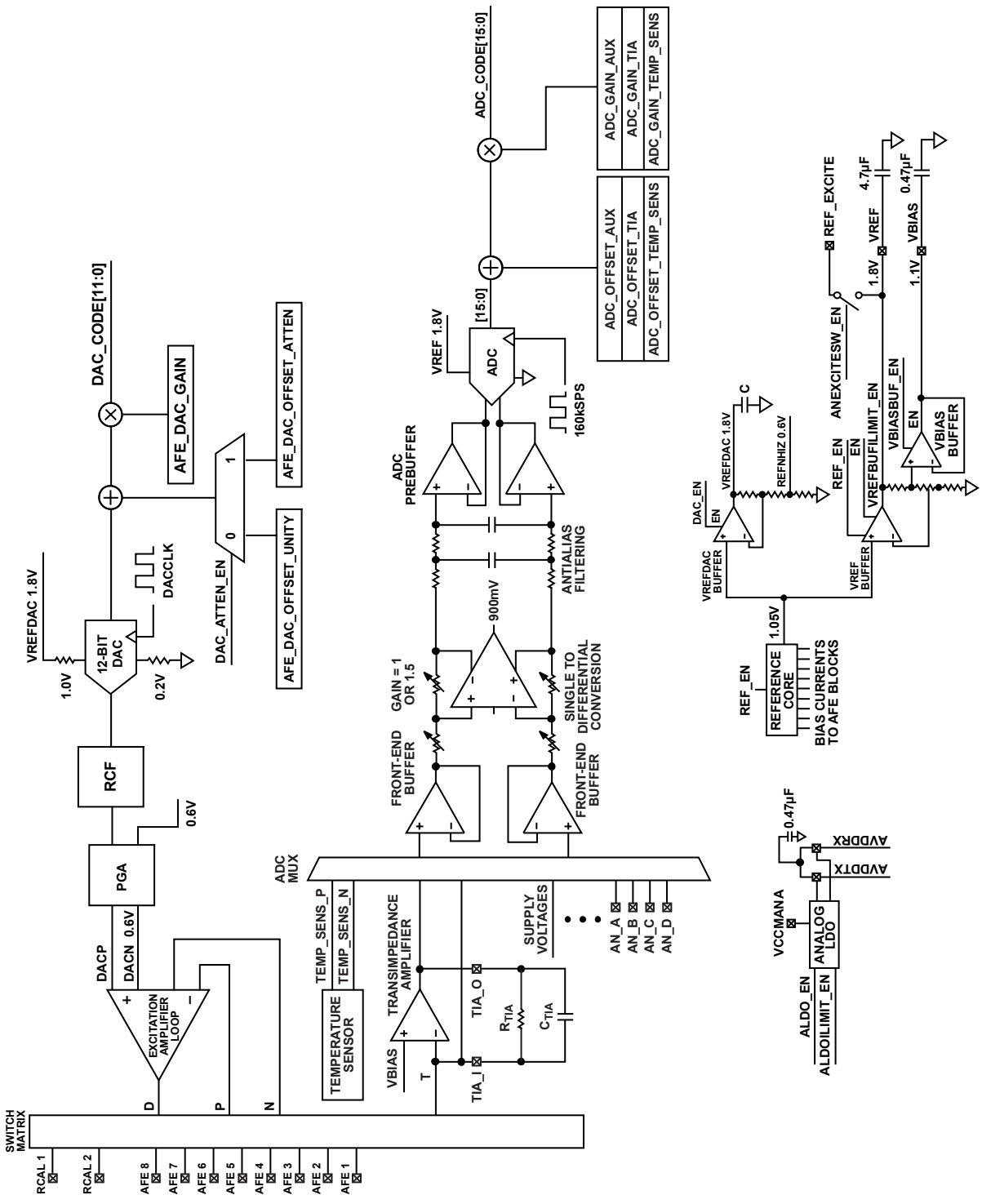


Figure 92. AFE Detailed Block Diagram

MODULE ARCHITECTURE

The module block diagram of the AFE is shown in Figure 93.

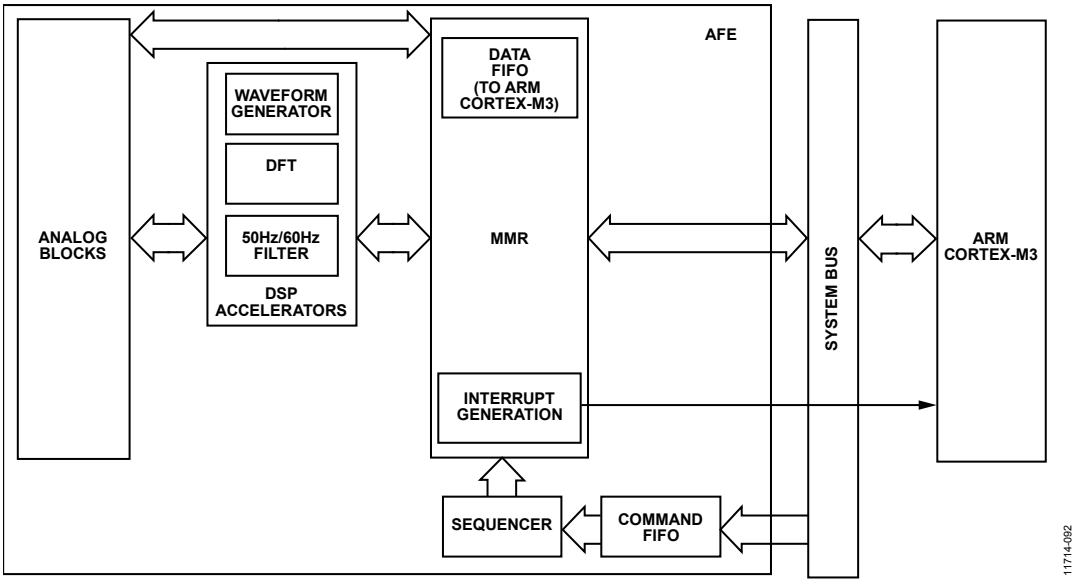


Figure 93. Module Block Diagram

The AFE can be controlled by Cortex-M3 via a collection of MMRs, which begin at Base Address 0x40080000. All blocks within the AFE are fully controllable and observable using the MMR registers. The MMRs can be accessed directly through the AHB or indirectly through the programmable sequencer.

The sequencer handles low level AFE operations and allows the AFE to perform its functions independent of the microcontroller.

A dedicated interrupt for the Cortex-M3 is generated by the AFE from a number of sources, including FIFOs and ADCs. All interrupts can be masked by the user.

Two DMA channels are available to the user: one for transferring data from the Cortex-M3 system to the AFE and another for transferring data from the AFE to the Cortex-M3 system. Using the first DMA channel, sequencer commands can be written to the AFE for later execution. The second DMA channel is dedicated to AFE results, such as ADC conversion results or DSP hardware accelerator outputs.

The custom DSP hardware accelerators implement operations commonly used in the ADuCM350 and include the following:

- A discrete Fourier transform (DFT) that is used for ac impedance measurements. It presents the Cortex-M3 with the real and imaginary parts of a complex result.
- A 50 Hz/60 Hz rejection filter—a low-pass filter used to filter out the supply interference in the dc measurement phase.
- A waveform generator that can output waveforms used for both ac and dc phases (sinusoids and trapezoids) without Cortex-M3 intervention. Arbitrary waveform generation is supported through using the sequencer with direct writes to the DAC.

The AFE controller operates on two clock domains:

- HCLK_BUS: used for data transfers over the AHB.
- ACLK: used for AFE state machines and sequencer.

HCLK_BUS and ACLK are synchronous with each other but are gated independently.

ACLK is further divided in the AFE to provide the DAC update clock and the ADC sampling clock.

Note that the AFE interface is designed for an ACLK frequency of 16 MHz. Changing the ACLK frequency divider in the CLKCON2 register is not supported.

OPERATION

The AFE memory mapped registers (MMRs) occupy addresses starting from 0x40080000 in the Cortex-M3 address space. A detailed description of the registers is provided in the AFE Memory Mapped Registers section.

Typically, the MMRs can be written from two sources: the Cortex-M3 and the AFE sequencer. The Cortex-M3 accesses the MMRs through an AHB slave interface. The sequencer uses a slice of the address (Bits[7:2]).

Reading back registers does not cause resource conflicts. To avoid write access conflicts, the AHB writes to MMRs are disabled while the sequencer is on (SEQ_EN = 1). There are exceptions to this rule as follows:

- The AFE_CMD_FIFO_WRITE register allows sequencer programming.
- The AFE_SEQ_CFG register allows ending the sequencer execution (SEQ_EN) and halting the sequence (SEQ_HALT).
- The AFE_x_x_INT registers allow clearing interrupts after servicing them.
- The AFE_x_x_IEN registers allow dynamic enabling/disabling of interrupts.

If the sequencer is on and the Cortex-M3 attempts to write to an MMR that is not in the exception list, an error is generated on the AHB (HRESP[1:0] = 01) and the write is ignored. This applies strictly to writes from the Cortex-M3 side; a sequencer write to an MMR does not trigger an error.

The same response is generated if writing to a read only register is attempted, regardless of the sequencer being enabled or disabled.

The following registers can be written to only if the sequencer is disabled (SEQ_EN = 0):

- AFE_ADC_RESULT
- AFE_DFT_RESULT_REAL
- AFE_DFT_RESULT_IMAG
- AFE_SUPPLY_LPF_RESULT
- AFE_TEMP_SENSOR_RESULT

An error is triggered if an attempt is made to write to these registers while the sequencer is enabled (SEQ_EN = 1). This is to facilitate user CRC checks by setting them to known values.

If the user attempts to read an unassigned address, the readback is 0x00000000 and an error is flagged (HRESP[1:0] = 01). Reading back unassigned or reserved bits reads back 0.

Programmable Sequencer

The role of the sequencer is to allow offloading of the low level AFE operations from the Cortex-M3 and provide cycle accurate control over the analog and DSP blocks. The sequencer handles timing critical operations without being subject to system load.

The AFE sequencer is clocked by ACLK, with a frequency of 16 MHz. It uses clock gating internally to minimize power consumption when idle.

The sequencer reads commands from the command FIFO and, depending on the command, either waits a given amount of time or writes a value to an MMR. The execution is sequential, with no branching. The sequencer cannot read MMR values or signals from the analog/DSP blocks. If a particular sequence needs conditional execution, it must be split between the sequencer and the Cortex-M3, with the decision block being executed on the Cortex-M3. In this case, the sequencer is programmed with commands up to the conditional point, and upon completion of those commands, a Cortex-M3 interrupt service routine determines the subsequent commands to be executed and proceed to load them into the command FIFO.

There are two types of commands that can be executed by the sequencer: write (MSB = 1) and timer (MSB = 0).

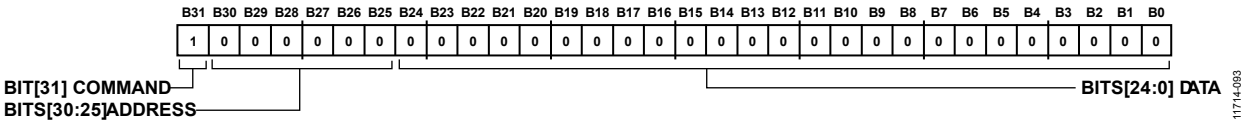


Figure 94. Write Command

The address field is 6 bits wide, allowing access to 64 MMRs. All MMR accesses are 32 bits only; byte and halfword accesses are not allowed. All accesses are implied write only. There is a direct mapping between the MMR address and the ADDRESS field used in the write command. ADDRESS corresponds to Bits[7:2] of the 32-bit MMR address.

For example, when the Cortex-M3 writes to the AFE_WG_CFG register, it uses the 0x40080014 address. To write to the same register using the sequencer, the address field must be 0b000101 (Bits[7:2] of the address used by the Cortex-M3).

The data field is 25 bits wide and allows writing to the MMR bits, Bits[24:0]. To keep the width of the command FIFO in line with the AHB, writing to the full 32 bits of the MMRs via the sequencer is not possible. However, because the MMR bits, Bits[31:25], are not used by any of the MMRs, all assigned MMR bits can be written by the sequencer.

There are two timer commands in the sequencer, with a separate hardware counter for each.

The wait command (see Figure 95) introduces wait states in the sequencer execution. After the programmed counter reaches 0, the execution is resumed by reading the next command from the FIFO.

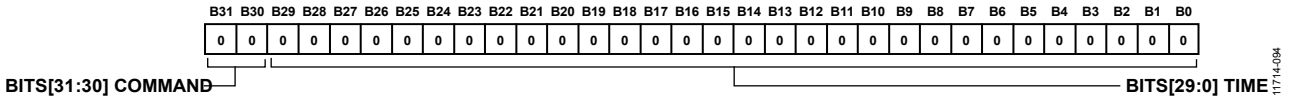


Figure 95. Wait Command

The timeout command (see Figure 96) starts a counter that works independent of the sequencer flow. The current value of the counter can be read by the Cortex-M3 at any time through the AFE_SEQ_TIMEOUT register. There are two interrupt bits associated with this command: SEQ_TIMEOUT_FINISHED and SEQ_TIMEOUT_ERR. SEQ_TIMEOUT_FINISHED is asserted at the end of the timeout period. SEQ_TIMEOUT_ERR is asserted if, at the end of the timeout period, the sequencer has not reached the end of execution (END_OF_SEQ = 0).

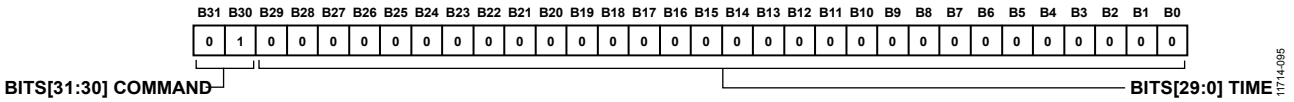


Figure 96. Timeout Command

The timeout counter is not reset when the sequencer execution is stopped as a result of a sequencer write command (writing a 0 to SEQ_EN). However, it is reset if the Cortex-M3 writes a 0 to SEQ_EN. This applies to situations when the Cortex-M3 must abort the sequence.

The time unit for both timer commands is one ACLK period. For a clock frequency of 16 MHz, the timer resolution is 62.5 ns, and the maximum timeout is ~67.1 sec. This is true even if the SEQ_WRITE_TIMER bits are nonzero.

While the MMRs, with the exception of calibration registers, are located in the address space that is writable by the sequencer, only a subset of these registers can be written by the sequencer. In addition to the read only registers, sequencer writes to the following registers have no effect:

- AFE_CMD_FIFO_WRITE
- AFE_ANALOG_CAPTURE_INT
- AFE_CMD_FIFO_INT
- AFE_DATA_FIFO_INT
- AFE_SEQ_COUNT
- AFE_SEQ_CRC

If the sequencer attempts to write to any of these registers, the write is silently ignored. If such a write was attempted due to an error (such as the wrong command being read from the memory), it is detected when checking the CRC and command count values.

A way to control the rate at which sequencer commands are executed is provided through the SEQ_WRITE_TIMER bits in the AFE_SEQ_CFG register. When a write command is executed by the sequencer, it performs the MMR write and then wait SEQ_WRITE_TIMER clock cycles before fetching the next command from the FIFO. The effect is the same as having a write command followed by a wait command. The main purpose is to reduce the code size when generating arbitrary waveforms. SEQ_WRITE_TIMER does not have any effect following a wait or timeout command.

In addition to a single write command being followed by a wait command, multiple write commands can be executed in succession followed by a wait command. Therefore, any configuration can be set up rapidly by the sequencer, regardless of the number of register writes followed by a precisely executed delay.

Typically, the sequencer is idle. Writing a 1 to the SEQ_EN bit in the AFE_SEQ_CFG register starts the sequencer. Only the Cortex-M3 can write SEQ_EN = 1, this being the only way to start the sequencer. The execution flow diagram is shown in Figure 97.

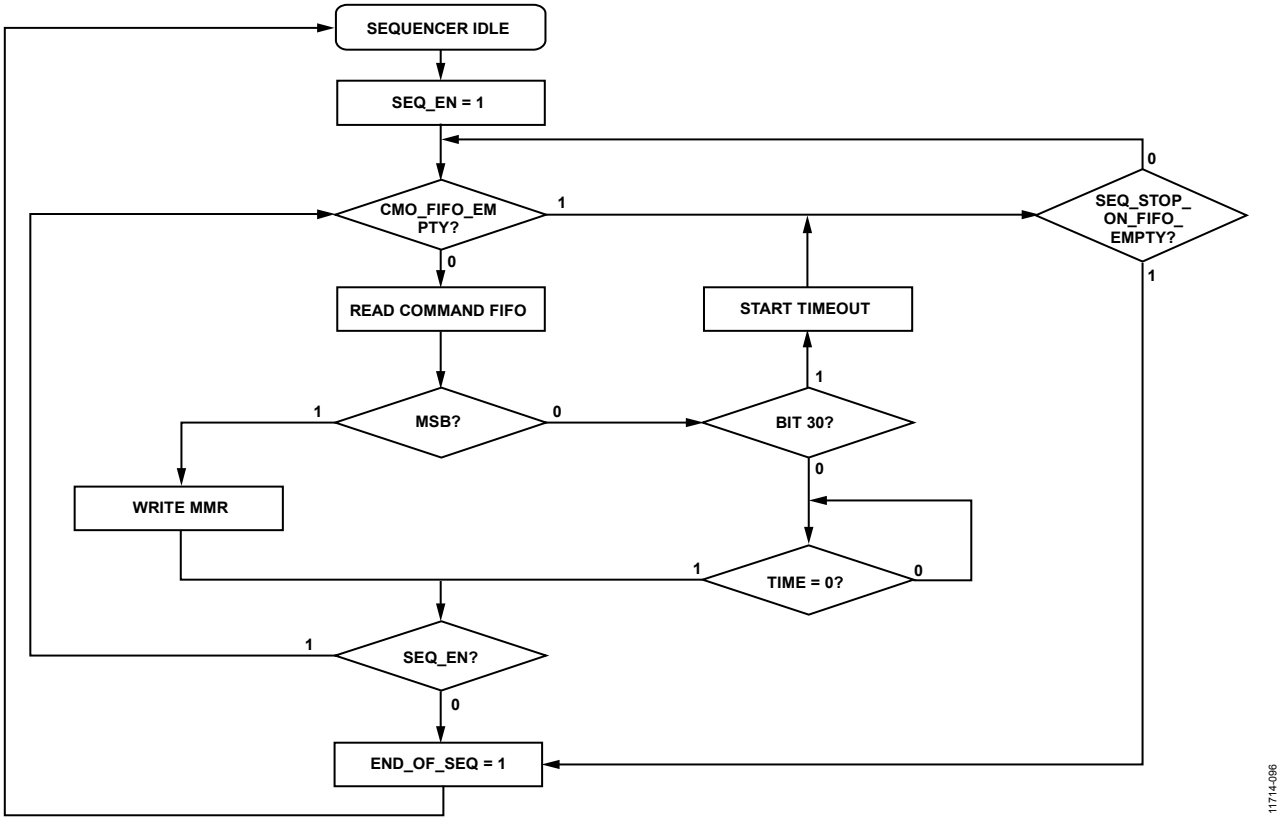


Figure 97. Sequencer Flow Diagram

The last command in any sequence writes a 0 to SEQ_EN. This forces the end of the execution and triggers the END_OF_SEQ interrupt.

If SEQ_STOP_ON_FIFO_EMPTY = 1, the sequencer is turned off if trying to read from command FIFO when empty. This is also a valid way to end the sequence. It may be useful to turn this flag off if the sequence has only a minimum timing specification and to turn this flag on for sequences with strict timing.

To abort a sequence, the Cortex-M3 must perform several writes to the AFE to disable the sequencer and stop the different blocks that may be used at this point. An example abort sequence is described in Table 468. The first two actions described (disable sequencer and disable command FIFO) always must be done. The rest of the actions may or may not be required, depending on the AFE mode of operation when aborting the sequence is needed.

Table 468. Abort Sequence Example

Action	Register	Bit(s)	Value	Comments
Disable Sequencer	AFE_SEQ_CFG	SEQ_EN	0	Resets both internal timers (for wait and timeout commands)
Disable Command FIFO	AFE_FIFO_CFG	CMD_FIFO_EN	0	Reset read/write pointers
		CMD_FIFO_DMA_REQ_EN	0	Disables command FIFO DMA requests
Disable DSP Blocks	AFE_CFG	WAVEGEN_EN	0	Disables waveform generator
		DFT_EN	0	Disables DFT engine
		SUPPLY_LPF_EN	0	Disables supply rejection filter
Stop Temperature Measurement	AFE_CFG	TEMP_CONV_EN	0	
Stop ADC Conversions	AFE_CFG	ADC_CONV_EN	0	
Power-Down Analog Blocks	AFE_CFG	As needed		

A way to pause the sequencer execution is provided through the SEQ_HALT bit in the AFE_SEQ_CFG register. When the bit is 1, the sequencer execution is stopped. This applies to every AFE function, including FIFO operations, internal timers, waveform generation, and data capture. Cortex-M3 reads from the MMRs are allowed, and this mode is intended to be used for debug purposes during software development.

The number of commands executed by the sequencer can be read from the AFE_SEQ_COUNT register. Each time a command is read from the FIFO and executed, the counter is incremented by 1. Performing a write to the AFE_SEQ_COUNT register resets the counter.

The sequencer calculates the CRC of all commands it executes. The algorithm used is CRC-8, using the polynomial $x^8 + x^2 + x + 1$.

The CRC-8 algorithm performs on 32-bit input data (sequencer instructions). Each 32-bit input is processed in one clock cycle, and the result is available immediately for reading by the Cortex-M3.

The CRC value can be read from the AFE_SEQ_CRC register. This register is reset by the same mechanism as the command count, by writing to the AFE_SEQ_COUNT register. The AFE_SEQ_CRC resets to a seed value of 0x01. Attempting to write to the AFE_SEQ_CRC register results in an error response that is the same as writing to any other read only register.

Command FIFO

The command FIFO is the link between the Cortex-M3 and the AFE sequencer. It has a depth of 8 words of 32 bits each. Each word is a sequencer command (write or wait). It is always unidirectional: the Cortex-M3 writes commands through the AFE_CMD_FIFO_ WRITE register, and the sequencer reads the commands.

The FIFO is synchronous: HCLK_BUS is used for writing, and ACLK is used for reading. Therefore, there is no need for synchronization.

A number of flags are associated with the command FIFO, which include the following:

- Empty (CMD_FIFO_EMPTY_STATUS): 1 if there are no words in the FIFO; 0 otherwise.
- Full (CMD_FIFO_FULL_STATUS): 1 if there are 8 words in the FIFO; 0 otherwise.
- Overflow (CMD_FIFO_OVF_STATUS): set to 1 for one clock period if the Cortex-M3 tries to write a word to the FIFO when the FIFO is full; 0 otherwise.
- Underflow (CMD_FIFO_UDF_STATUS): set to 1 for one clock period if the sequencer tries to read from an empty FIFO; 0 otherwise.
- Threshold (CMD_FIFO_THR_STATUS): 1 if the number of words in the FIFO is below the threshold set in CMD_FIFO_THR_VAL (the AFE_FIFO_CFG register); 0 otherwise. This flag is also 0 if the FIFO is empty.

The flags are user readable through the AFE_STATUS register. Each of the flags has a maskable interrupt associated with it.

The overflow (CMD_FIFO_OVF_STATUS) and underflow (CMD_FIFO_UDF_STATUS) flags only activate for one clock period. However, the state of their corresponding interrupt bits (CMD_FIFO_OVF/CMD_FIFO_UDF) indicate if an overflow or underflow has occurred at any point during the FIFO operation.

The CMD_FIFO_EN bit in the AFE_FIFO_CFG register enables the FIFO when a 1 is written to it. If CMD_FIFO_EN = 0, the FIFO is reset by resetting the read and write pointers back to their default values (0 for both). The FIFO stays in reset mode until a 1 is written to CMD_FIFO_EN.

There is a DMA channel associated with the FIFO. To enable it, a 1 must be written to the CMD_FIFO_DMA_REQ_EN bit in the AFE_FIFO_CFG register. The DMA channel is disabled by default, due to CMD_FIFO_EN being 0 even though CMD_FIFO_DMA_REQ_EN is 1. When the FIFO is enabled (CMD_FIFO_EN = 1) and CMD_FIFO_DMA_REQ_EN = 1, the FIFO issues a DMA request any time it is not full.

The DMA request is deasserted when the FIFO is full. The DMA request is also disabled when the DMA controller has finished the programmed transfer. That means the current sequence execution is complete.

The number of words in the FIFO can be inspected at any time by reading the CMD_FIFO_WORD_COUNT bits in the AFE_STATUS register.

To avoid bottleneck issues, the command FIFO and its DMA channel must be enabled (CMD_FIFO_EN = 1, CMD_FIFO_DMA_REQ_EN = 1) before the sequencer (SEQ_EN = 1) is enabled. This fills up the FIFO before the sequencer begins reading commands.

Reading back from the command FIFO returns 0x00000000.

Interrupts

The AFE generates four interrupts for the NVIC. There several interrupt sources, such as FIFOs and analog/DSP blocks, that are arranged into groups; each group has an interrupt associated with it. All interrupts are maskable by the user.

The split of interrupts into groups is detailed in Table 469. More details about each interrupt can be found in the register map description in the AFE Memory Mapped Registers section.

Table 469. Interrupt Groups

Interrupt	Interrupt Sources
AFE_ANALOG_CAPTURE_INT (AFE_ANALOG_CAPTURE_IEN)	ADC_RESULT_READY (ADC_RESULT_READY_IEN) DFT_RESULT_READY (DFT_RESULT_READY_IEN) SUPPLY_LPF_RESULT_READY (SUPPLY_LPF_RESULT_READY_IEN) TEMP_RESULT_READY (TEMP_RESULT_READY_IEN)
AFE_ANALOG_GEN_INT (AFE_ANALOG_GEN_IEN)	DELAY_COMMAND_DONE (DELAY_COMMAND_DONE_IEN) HARDWARE_SETUP_DONE (HARDWARE_SETUP_DONE_IEN) BREAK_SEQUENCE_ORG (BREAK_SEQUENCE_ORG_IEN) CUSTOM_INT (CUSTOM_INT_IEN)
AFE_CMD_FIFO_INT (AFE_CMD_FIFO_IEN)	END_OF_SEQ (END_OF_SEQ_IEN) SEQ_TIMEOUT_FINISHED (SEQ_TIMEOUT_FINISHED_IEN) SEQ_TIMEOUT_ERR (SEQ_TIMEOUT_ERR_IEN) CMD_FIFO_FULL (CMD_FIFO_FULL_IEN) CMD_FIFO_EMPTY (CMD_FIFO_EMPTY_IEN) CMD_FIFO_THR (CMD_FIFO_THR_IEN) CMD_FIFO_OVF (CMD_FIFO_OVF_IEN) CMD_FIFO_UDF (CMD_FIFO_UDF_IEN)
AFE_DATA_FIFO_INT (AFE_DATA_FIFO_IEN)	DATA_FIFO_FULL (DATA_FIFO_FULL_IEN) DATA_FIFO_EMPTY (DATA_FIFO_EMPTY_IEN) DATA_FIFO_THR (DATA_FIFO_THR_IEN) DATA_FIFO_OVF (DATA_FIFO_OVF_IEN) DATA_FIFO_UDF (DATA_FIFO_UDF_IEN)

Each interrupt has two bits associated with it, each in a different register.

- AFE_x_INT/AFE_x_x_INT: latched interrupt. This bit is used to generate the interrupt for the NVIC (if unmasked). After being set, the bit stays set until the user clears it by writing a 1. Reading back a 1 from this bit means the interrupt has been triggered.
- AFE_x_IEN/AFE_x_x_IEN: interrupt enable. A 1 means the respective interrupt is used to generate the interrupt for the NVIC, and a 0 means the interrupt is ignored.

Additionally, for some of the interrupt sources, the current state of an interrupt source can be read through corresponding bits in the AFE_STATUS registers.

Each interrupt has an AFE_INT register and an AFE_IEN register associated with it, as shown in Table 469.

The interrupts in the AFE_ANALOG_GEN_INT group are a special case. They have no interrupt sources associated with them and are instead set by sequencer commands. The purpose of these interrupts is to introduce software hooks into a sequence and, therefore, allow the Cortex-M3 to perform additional steps (for example, to read the register map following a HARDWARE_SETUP_DONE interrupt).

The behavior of these bits when writing to them depends on the source.

- Source is Cortex-M3: same as the other interrupts (RW1C).
- Source is sequencer: writing a 1 sets the interrupt; writing a 0 has no effect. The bits in the AFE_x_x_INT registers are always set to 1 if an interrupt request has been raised by the source. However, this is not seen by the NVIC if the corresponding mask is not set to 1.

Figure 98 shows an example of how the interrupt generation is performed.

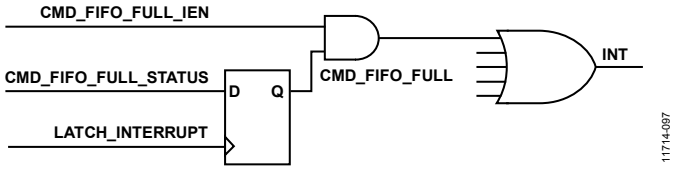


Figure 98. Interrupt Generation (LATCH_INTERRUPT is an Internal Signal)

Switch Matrix Control

There are two ways to control the switch matrix

- Encoded state control register: the full 35-bit switch state value is encoded using 17 bits (the AFE_SW_CFG register). However, only a subset of the total switch matrix combinations can be generated this way.
- Individual control: each switch is controlled independently. For this type of control, 35 bits are needed. These bits are located in the AFE_SW_FULL_CFG_MSB and AFE_SW_FULL_CFG_LSB registers.

Selecting the current source for switch control is performed using the SW_SOURCE_SEL bit (in the AFE_SW_CFG register). A write to SW_SOURCE_SEL is always required to update the switch state. When AFE_SW_CFG is used, a 0 is written to SW_SOURCE_SEL in the same MMR write. When AFE_SW_FULL_CFG_MSB/AFE_SW_FULL_CFG_LSB are used, a write to AFE_SW_CFG with SW_SOURCE_SEL = 1 must be performed before the switch state changes.

Because the switches are located in the VCCM domain, the control signals must be driven while the device is in hibernate mode. When the AFE is powered up, the switch control signals remain unchanged until a write to SW_SOURCE_SEL is performed. The switch control signals can be read through the AFE_SW_STATUS_MSB and AFE_SW_STATUS_LSB registers.

A block diagram of the switch control scheme is shown in Figure 99.

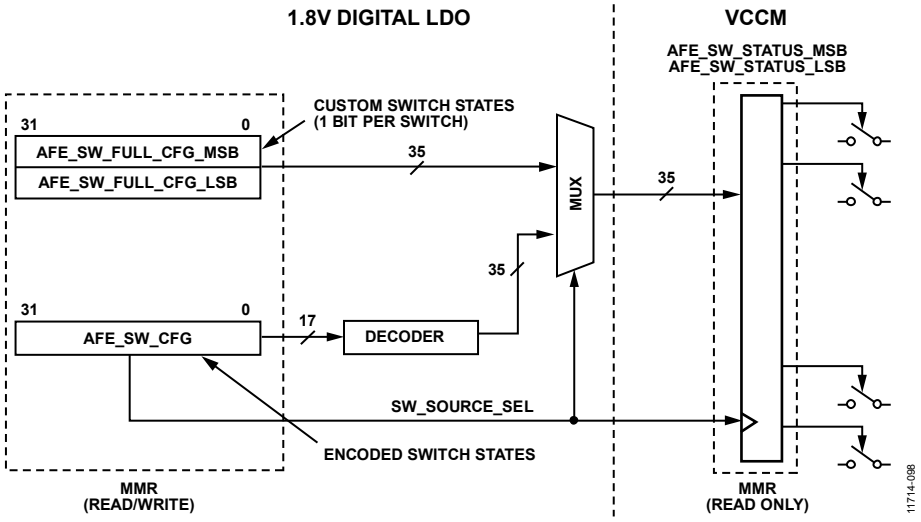


Figure 99. Switch Control Block Diagram

The 17-bit switch state representation is outlined in Figure 100. A blank location means the respective switch is open, and an x denotes that the respective switch is closed.

EXCITATION LOOP DRIVER

D MUX STATE	DR1	D2	D3	D4	D5	D6	D7	D8
0								
1	x							
2		x						
3			x					
4				x				
5					x			
6						x		
7							x	
8								x
9	x	x	x	x	x	x	x	x
10 TO 15								

POSITIVE INPUT IN-AMP

P MUX STATE	PR1	P2	P3	P4	P5	P6	P7	P8	PL
0									x
1	x								
2		x							
3			x						
4				x					
5					x				
6						x			
7							x		
8								x	
9 TO 14									x
15									

NEGATIVE INPUT IN-AMP

N MUX STATE	N1	N2	N3	N4	N5	N6	N7	NR2	NL
0									x
1	x								
2		x							
3			x						
4				x					
5					x				
6						x			
7							x		
8								x	
9 TO 14									x
15									

TRANSIMPEDANCE AMP INPUT

T MUX STATE	T1	T2	T3	T4	T5	T6	T7	TR2
0								
1	x							
2		x						
3			x					
4				x				
5					x			
6						x		
7							x	
8								x
9	x	x	x	x	x	x	x	x
10 TO 15								

11714-099

Figure 100. Switch Group Multiplexing

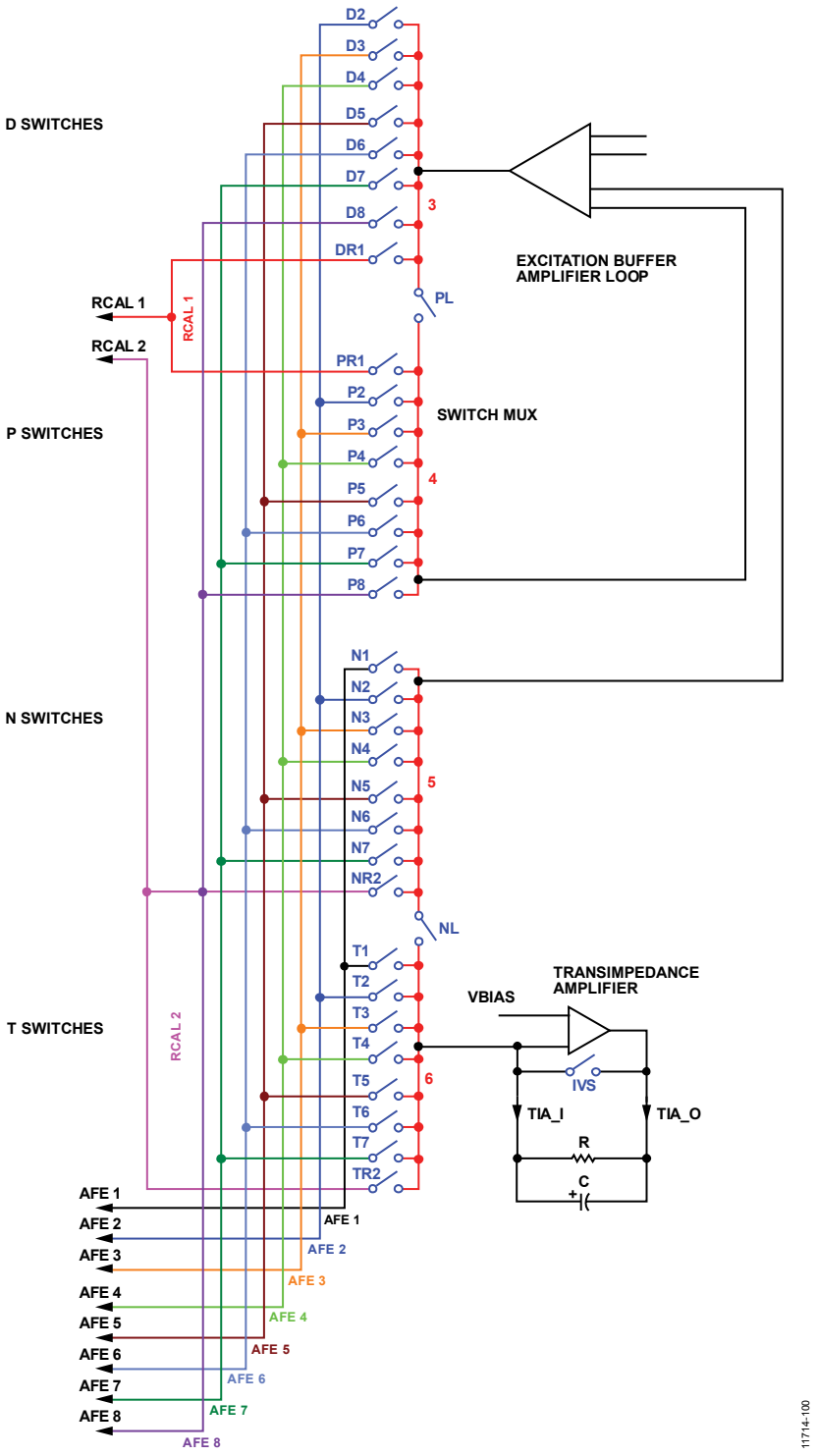


Figure 101. Switch Matrix

AFE Retention In Hibernate

The following list details what AFE registers/bits retain state in hibernate mode:

- AFE_DAC_OFFSET_UNITY
- AFE_DAC_OFFSET_ATTEN
- AFE_DAC_GAIN
- AFE_REF_TRIM0
- AFE_REF_TRIM1
- AFE_ALDO_TRIM
- AFE_EXBUF_TRIM
- AFE_TEMP_SENS_TRIM

WAVEFORM GENERATOR

The waveform generator block is responsible for creating parameterized waveforms needed to excite the external sensor in the ADuCM350. The waveforms are either sinusoids or trapezoids. Direct writes to the DAC are also possible. Arbitrary waveforms can be generated using the AFE sequencer and DMA transfers.

A simplified block diagram of the waveform generator is shown in Figure 102.

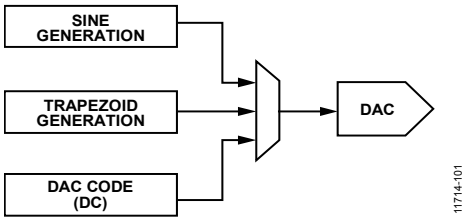


Figure 102. Waveform Generator Block Diagram

Bit 14 (WAVEGEN_EN) in the AFE_CFG register provides a global enable for the waveform generator block. When WAVEGEN_EN = 1, the selected source is started and keeps looping until either the block is disabled or another source is selected. When the block is disabled, the DAC output keeps its voltage until a different waveform is selected by writing to TYPE_SEL or the waveform is reset (for example, by writing a 1 to WG_TRAP_RESET).

Bit 0 (WG_TRAP_RESET) in the AFE_WG_CFG register provides a way to reset the trapezoid generator. This means resetting the current pointer to the start of Delay 1. The sinusoid generator is always reset when enabled.

Only the following two programmable input parameters require a WAVEGEN restart:

- AFE_WG_AMPLITUDE: the amplitude of the waveform.
- AFE_WG_PHASE: the phase offset of the waveform.

All of the other parameters (AFE_WG_DCLEVELx, AFE_WG_DELAYx, AFE_WG_SLOPEx, AFE_WG_FCW, and AFE_WG_OFFSET) take effect while the WAVEGEN_EN = 1.

In the case of successive DAC writes, care must be taken in allowing sufficient analog settling time of the DAC output. See the DAC Settling plot in the ADuCM350 data sheet.

Another consideration is that the DAC controller effectively ignores a successive DAC write if it occurs within 13 ACLK cycles (0.8125 µs) of the previous DAC write. This is due to the DAC gain/offset correction, which does not restart for the second DAC write, thus ignoring the second DAC write. This time is much smaller than the actual DAC settling time so it does not constrain the excitation stage.

Sinusoid Generator

The block diagram of the sinusoid generator is shown in Figure 103.

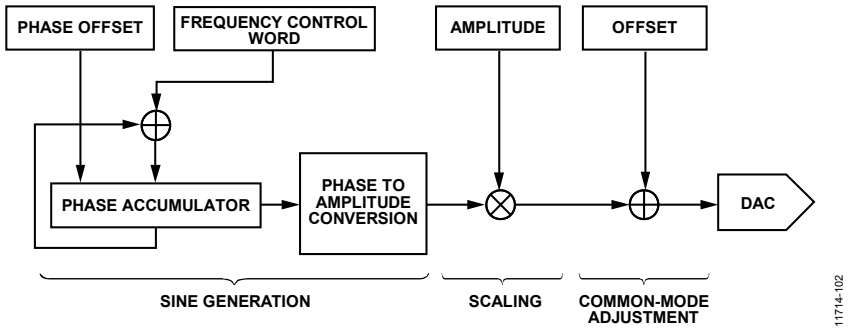


Figure 103. Sinusoid Generator Block Diagram

The output frequency is adjusted using the frequency control word (the SINE_FCW bit in the AFE_WG_FCW register) according to the following formula:

$$f_{OUT} = f_{ACLK} \times FCW/2^{26}$$

where f_{ACLK} is the frequency of ACLK (16 MHz).

The frequency control word width is 26 bits, which translates to an output frequency resolution of ~0.24 Hz in the worst case scenario (f_{ACLK} = 16 MHz).

The sinusoid generator includes programmable phase offset (the SINE_PHASE_OFFSET bit in the AFE_WG_PHASE register). When enabled (WAVEGEN_EN = 1 and TYPE_SEL = 10), the phase accumulator is initialized with the contents of the phase offset register. After the sinusoid generator is started, the phase increment is always positive.

The phase is converted to amplitude using a lookup table.

The sinusoid generator maintains its output when disabled and resets to the SINE_PHASE_OFFSET value when enabled. Programmable amplitude (the SINE_AMPLITUDE bit of the AFE_WG_AMPLITUDE register) and offset (the SINE_OFFSET bit of the AFE_WG_ OFFSET register) values are used for scaling and common-mode adjustment of the sinusoid output. Their effect is shown in Figure 104.

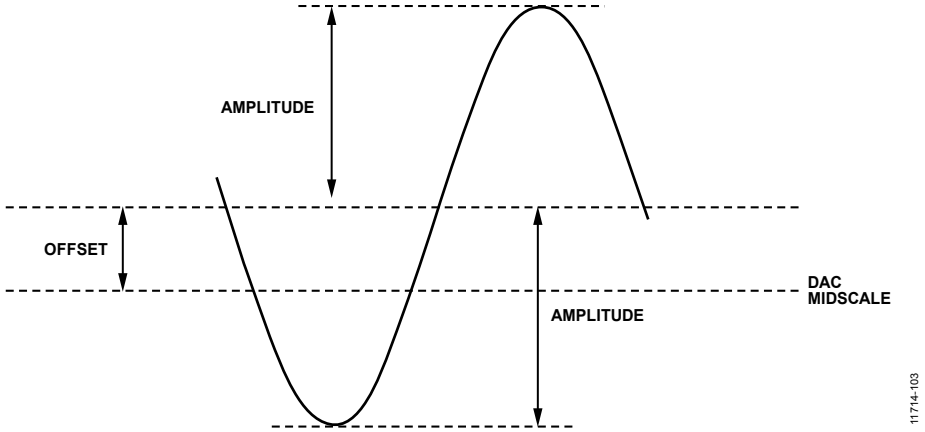


Figure 104. Sine Wave Amplitude and Offset Adjustment

An important feature of the AFE is the ability to stop the output waveform at a desired voltage (typically midscale). The AFE sequencer can be used to do this by using a wait command and then writing a 0 to WAVEGEN_EN.

Ignoring all the other AFE blocks, the following example sequence generates one period of 20 kHz sinusoid (where the sequencer clock is 16 MHz, and one period of the output signal is 800 sequencer clocks):

```
// Select sinusoid generator
0x8A000034
// Enable waveform generator (ignoring the other enable
//    signals in AFE_CFG register)
0x80004000
// Wait 800 clock periods
0x0000031E
// Disable waveform generator
0x80000000
```

In regards to the wait 800 clock periods within this example sequence, note that to get a true wait of 800 clock cycles between turning the waveform generator on and off, a wait of 798 (0x31E) must be used. (One clock cycle can be subtracted because of the wait command, and another clock cycle can be subtracted because of the enable/disable command.)

Refer to the Programmable Sequencer section under the write command for more information. This sequence can be programmed by the Cortex-M3 directly or by DMA transfers that involve a write to the AFE_CMD_FIFO_WRITE register.

Trapezoid Generator

The definition of the trapezoid waveform is shown in Figure 105.

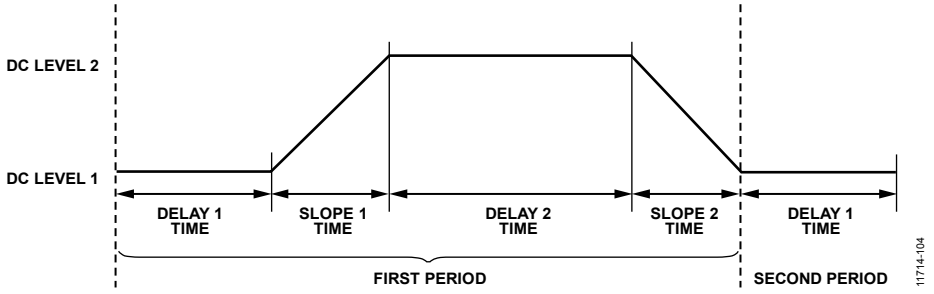


Figure 105. Trapezoid Waveform Definition

The six variables shown in Figure 105 are user programmable through the AFE_WG_DCLEVEL_1, AFE_WG_DCLEVEL_2, AFE_WG_DELAY_1, AFE_WG_SLOPE_1, AFE_WG_DELAY_2, and AFE_WG_SLOPE_2 registers. These variables, together, define the trapezoid waveform. The times are expressed in number of periods of the DAC update clock.

A maximum delay time for AFE_WG_SLOPE register, Bits [18:0], is defined by the following equation:

$$Slope\ Delay = (2^{19} - 1) \times \frac{\left(\frac{1}{16 \times 10^6}\right)}{50} = 1.638\ \text{sec}$$

A period of the trapezoidal waveform starts at the beginning of Delay 1 and finishes at the end of Slope 2. When a period finishes, it starts from the beginning of the period again. The waveform generator keeps looping until it is disabled by the user.

There is no requirement for DC Level 1 to be less than DC Level 2; the engine checks the sign of the change and reverse the slope direction if needed.

The step size for either slope is recalculated at the beginning of the slope to allow dynamic change of the trapezoid waveform using the sequencer. The step size is fixed for the duration of the slope. Internally, the step size is calculated with fractional precision, and the new code is rounded before it is written to the DAC.

The following cases are examples of how stepping through the slope works. The DAC update rate is assumed to be 320 kHz, the slope duration is 31.25 µs, and the slope starts at DAC Code 100 (decimal). Three cases are considered for the slope end: DAC Code 125 (Case 1), DAC Code 124 (Case 2), and DAC Code 120 (Case 3).

For Case 1, the step size is 2.5; for Case 2, the step size is 2.4; and for Case 3, the step size is 2. Table 470 shows the DAC code for each step.

Table 470. Example of Stepping Through the Slope

Step Number	DAC Code (Case 1)	DAC Code (Case 2)	DAC Code (Case 3)
1	103	102	102
2	105	105	104
3	108	107	106
4	110	110	108
5	113	112	110
6	115	114	112
7	118	117	114
8	120	119	116
9	123	122	118
10	125	124	120

In Case 2, the step size of 2.4 has an infinite number of fractional bits. However, the step size is stored internally with enough precision that the difference between the ideal and actual DAC codes is less than or equal to 1. A flow diagram for the trapezoid generation is shown in Figure 106.

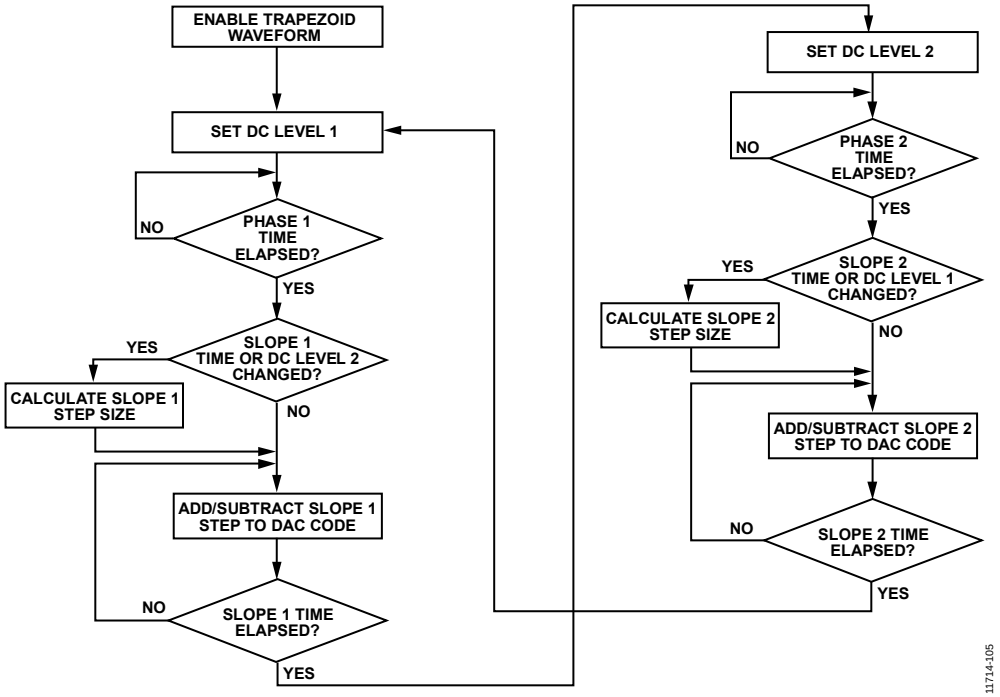


Figure 106. Flow Diagram for Trapezoid Generation

For example, generation of the dc waveform is shown in Figure 107, with details shown in Figure 108 and Figure 109.

The DAC update rate is fixed at 320 kHz.

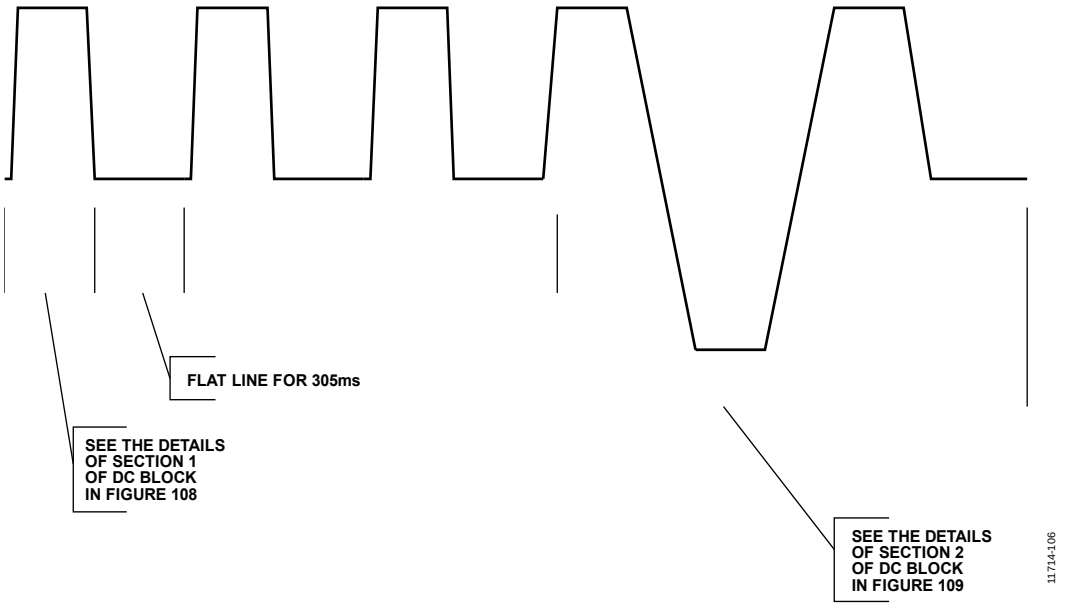


Figure 107. DC Block

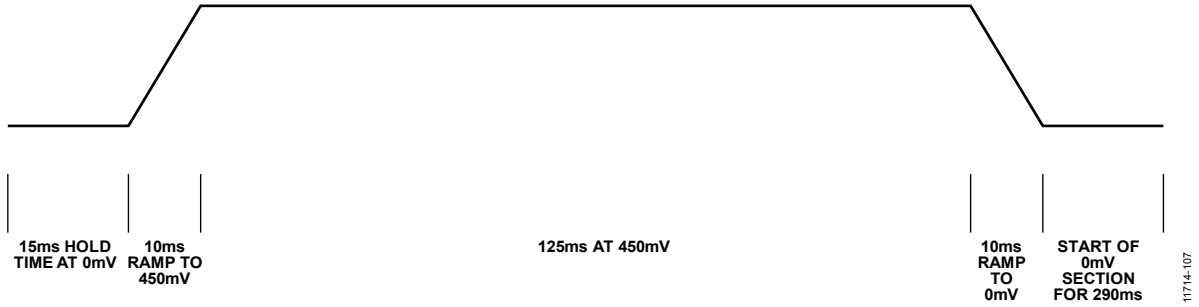


Figure 108. DC Block, Section 1

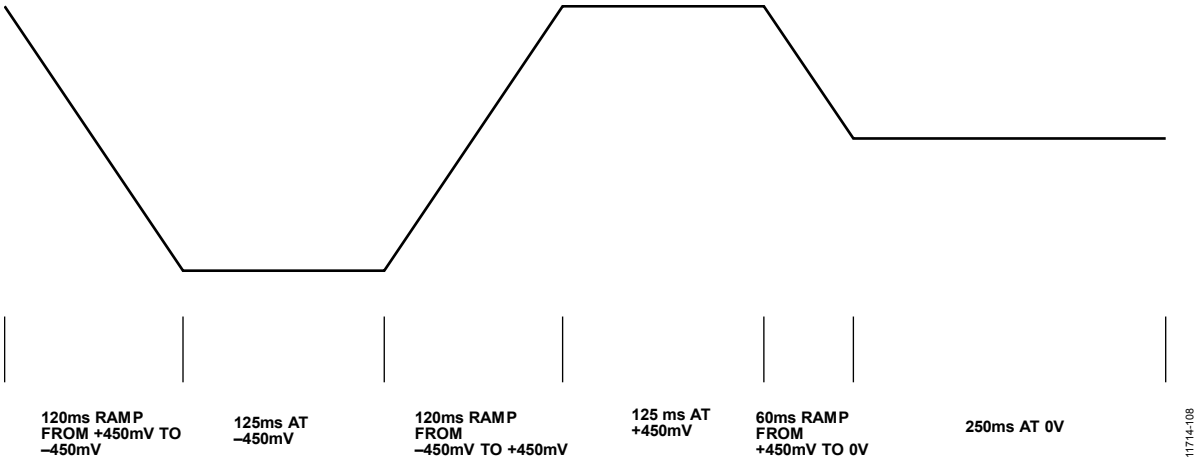


Figure 109. DC Block, Section 2

The following describes how to generate the defined code for the sequencer to generate the trapezoid waveform:

- DC level.
  - DC Level 2: The MSB is set to indicate a register write, and the next six bits are Bits[7:2] of the DC Level 2 MMR address.
  - For Address 0x4008001C, this is 0001110; therefore, 1000 1110 = 8E.
  - For Section 1, DC Level 1 is 0 mV and DC Level 2 is 450 mV. This corresponds to DAC Code 0x800 and DAC Code 0xC80, respectively.
  - The DAC range is −800 mV to +800 mV. The LSB size is 1.6 V/4095 = 390.720 μV; therefore, 450 mV = 1152 LSBs, which is 0x480.
  - Offsetting this from midscale (0 V, 0x800) results in a DAC code of 0xC80.
  - The resulting 32-bit command word to set DC Level 2 is 0x8E000C80 for DC Level 2 = 450 mV.
- Delay.
  - The MSB is set to indicate a register write, and the next six bits are Bits[7:2] of the DC Level 2 MMR address.
  - For Address 0x40080020, this is 0010000; therefore, 1001 0000 = 90.
  - The DAC update rate is 320 kHz, resulting in a period of 3.125 μs, which leads to 4800 periods for 15 ms, or 0x12C0.
  - The resulting 32-bit command word is 0x900012C0 for Delay 1 = 15 ms.

Section 1 starts with Delay 1 Time = 15 ms and Delay 2 Time = 125 ms. The register values for these are 0x012C0 and 0x09C40, respectively.

Starting with the second period, Delay 1 Time is 305 ms (Register Value 0x17D40). The new value must be written any time between 15 ms and 160 ms, from the beginning of the waveform, to be valid for the start of the second period. A value of 100 ms has been chosen, which corresponds to a timer value of 0x186A00 (sequencer timer, clocked by a 16 MHz clock).

Both slopes are 10 ms, Register Value 0x00C80.

Based on the this information, the sequence to generate DC Section 1 is (ignoring all other AFE blocks) as follows:

```
// Set DC Level 1 to 0 mV
0x8C000800
// Set DC Level 2 to 450 mV
0x8E000C80
// Set Delay 1 Time to 15 ms
0x900012C0
// Set Slope 1 Time to 10 ms
0x92000C80
// Set Delay 2 Time to 125 ms
0x94009C40
// Set Slope 2 Time to 10 ms
0x96000C80
// Enable waveform generator
0x80004000
// Wait 100 ms
0x00186A00
// Set Delay 1 Time to 305 ms
0x90017D40
```

Refer to the Programmable Sequencer section under the write command. This sequence can be programmed by the Cortex-M3 directly or by DMA transfers that involve a write to the AFE_CMD_FIFO_WRITE register.

The previous sequencer generates the full Section 1 of the dc block. The first six commands are setup commands that are not timing critical. These commands can be written directly by the Cortex-M3.

From the end of the previous sequence until the start of DC Block Section 2 in Figure 109, there is an additional delay of 1275 ms.

```
// Wait 1275 ms
0x01374780
```

For DC Block Section 2, the first segment is a dc phase of 450 mV. In Section 1, this was DC Level 2; therefore, DC Level 1 is from 0 mV to −450 mV. This corresponds to a DAC code of 0x380.

The first two slopes are now 120 ms (Register Value 0x9600), with the last slope equal to 60 ms (0x4B00). The 60 ms slope must be programmed at a time between 125 ms and 615 ms from the beginning of Section 2. A value of 600 ms has been chosen (Timer Value 0x927C00), which puts the change during the last 450 mV phase.

Both delay times are now 125 ms; therefore, the Delay 1 time must change to 125 ms. The last delay time is a Delay 1 time of 250 ms (0x13880) at 0 mV.

The full duration of Section 2 is 925 ms. After subtracting the 600 ms already timed, a wait time of 325 ms (0x4F5880) is required until the end of the dc block.

The sequence that implements the dc block of Section 2 is as follows:

```
// Set DC Level 1 to -450 mV
0x90000380
// Set Slope 1 Time to 120 ms
0x92009600
// Set Slope 2 Time to 120 ms
0x96009600
// Wait 600 ms
0x00927C00
// Set Slope 2 Time to 60 ms
0x96004B00
// Set DC Level 1 to 0 mV
0x8C000800
// Set Delay 1 time to 250 ms
0x90013880
// Wait 325 ms
0x004F5880
// Disable waveform generator
0x80000000
```

Updating Trapezoid When Enabled

To ensure that the timing parameters are being applied to the next segment, the MMR writes must take place at least 50 ACLK cycles before the segment start. If the next segment is a slope, there are additional 50 ACLK cycles required, due to calculating the slope increment through sequential division.

A safe way to update the timing parameters is as follows:

- TRAP_DELAY_1: update at the start of Delay 2 segment.
- TRAP_DELAY_2: update at the start of Delay 1 segment.
- TRAP_SLOPE_1: update at the start of Slope 2 segment.
- TRAP_SLOPE_2: update at the start of Slope 1 segment.

The trapezoid output is continuously compared with the bounds (DC_LEVEL_1 and DC_LEVEL_2). If outside bounds, the trapezoid output is clamped at either DC_LEVEL_1 or DC_LEVEL_2, depending on the dc values. During Delay 1/Delay 2 segments, the trapezoid output is always within bounds.

It is also possible to update the dc levels during slope segments. However, avoid a change in bounds that causes the output to clamp.

A safe way to update the dc levels is as follows:

- DC_LEVEL_1: update at start of Delay 2 segment, or 50 ACLK cycles before the start of Delay 2 segment if TRAP_DELAY_2 < 2.
- DC_LEVEL_2: update at start of Delay 1 segment, or 50 ACLK cycles before the start of Delay 1 segment if TRAP_DELAY_1 < 2.

An additional 50 ACLK cycles are needed to ensure that the slope is calculated using the new dc level if the current waveform is a triangular waveform.

The trapezoid generator requires 1 DAC update period (50 ACLK cycles); therefore, a TRAP_DELAY value of 0 has the same effect on the output waveform as a value of 1. This is not an offset of 50 ACLK cycles for each TRAP_DELAY; it is only applicable when TRAP_DELAY = 0.

ADC RECEIVE STAGE

The output data rate of the ADC depends on which point in the chain is being interrogated. There are two points at which the data can be read.

- 160 kSPS at the output of the gain correction stage.
- 900 SPS at the output of the power line reject filter.

The ADC data can be interrogated using any of the following methods:

- By raw data
- Through a DFT engine
- Through a 50 Hz/60 Hz supply rejection filter

Discrete Fourier Transform

This block performs a 2048-point single frequency DFT. It takes the 16-bit ADC output as input and outputs the real and imaginary parts of the complex result.

The DFT engine calculates the signal power at a single frequency bin, which is the bin corresponding to the excitation frequency. As such, there is a tight coupling between the sinusoid waveform generator and the DFT engine. The frequency control word (AFE_WG_FCW) is used by the DFT engine to determine the required frequency bin.

The DFT engine outputs the results as a complex number. The magnitude and phase of the impedance at the excitation frequency are calculated by the Cortex-M3 using the following formulas:

$$Magnitude = \sqrt{(R^2 + I^2)} \text{ and } Phase = \text{atan}(I/R)$$

where:
R is the real part, read from the AFE_DFT_RESULT_REAL register.
I is the imaginary part, read from the AFE_DFT_RESULT_IMAG register.

When the DFT engine is enabled (DFT_EN = 1 of the AFE_CFG register), it monitors the ADC output. When a new result is available, the DFT engine reads the result and uses it to calculate the next DFT term.

Both the sinusoid generator and the DFT engine are based on the same phase accumulator, which allows the DFT to be started at any time, n, and to generate a magnitude and phase that are phase shifted from the current sinusoid generator phase.

The ADC data is first windowed, using a Hann window function. This is needed to minimize spectral leakage because the system is not guaranteed to be coherent.

The DFT equation is applied to the windowed data and has the following format:

$$R = \sum_{n}^{n+2047} x(i)\cos(2 \times \pi \times i \times \frac{f_{OUT}}{f_{ADC}})$$

where *n* is the phase when the DFT engine starts sampling.

$$I = \sum_{n}^{n+2047} x(i)\sin(2 \times \pi \times i \times \frac{f_{OUT}}{f_{ADC}})$$

After 2048 samples have been read and processed, the DFT raises the DFT_RESULT_READY flag, which triggers the DFT_RESULT_READY interrupt. The DFT results are available for reading from the AFE_DFT_RESULT_REAL and AFE_DFT_RESULT_IMAG registers. After 2048 samples have been processed and if DFT_EN stays 1, the DFT engine restarts itself and produces another result after 2048 additional samples. To stop it, DFT_EN = 0 must be written into AFE_CFG.

Each time the DFT engine is started (DFT_EN = 1), the internal registers are reset and the DFT calculation restarts. The result registers (AFE_DFT_RESULT_REAL and AFE_DFT_RESULT_IMAG) state is kept until a 2048-point DFT calculation is complete, and the new real and imaginary results are written to the result registers.

A way to accommodate noncoherent sampling is provided by changing the divide ratio for the DAC clock. The following explains the motivation behind noncoherent sampling.

A coherently sampled system is shown in Figure 110. In this example, the DAC update rate is an integer multiple of the ADC sampling rate. The image of the DAC output frequency is aliased back in band after being attenuated by the reconstruction filter and the antialiasing filter. This contributes to the total measurement error.

Figure 111 provides another way to look at the coherent sampling. The top plot shows the DAC output spectrum (up to 1.28 MHz), and the bottom plot shows the FFT of the ADC output (0 kHz to 160 kHz). In this example, both the DAC and the ADC are clocked at 160 kHz. In addition, all the images of the excitation frequency are aliased back in the same FFT bin as the fundamental.

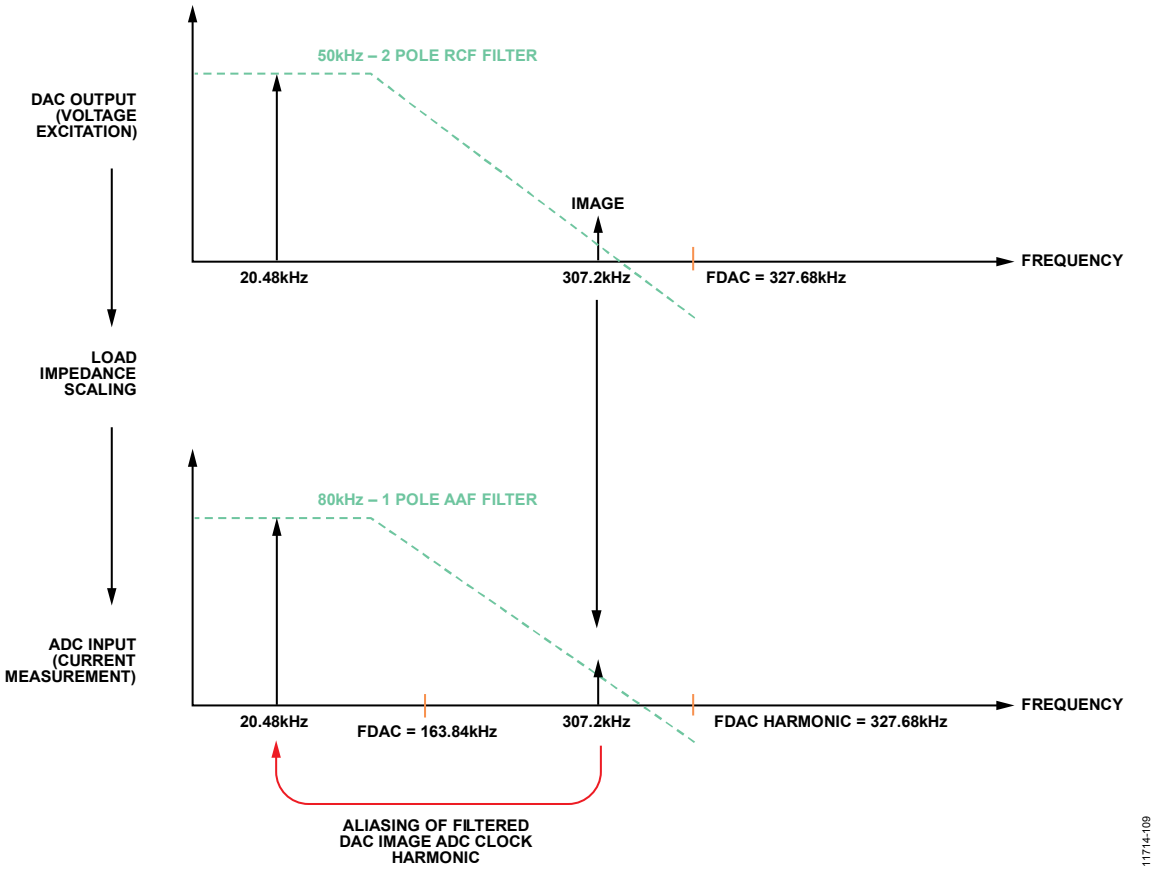


Figure 110. Image Aliasing In-Band

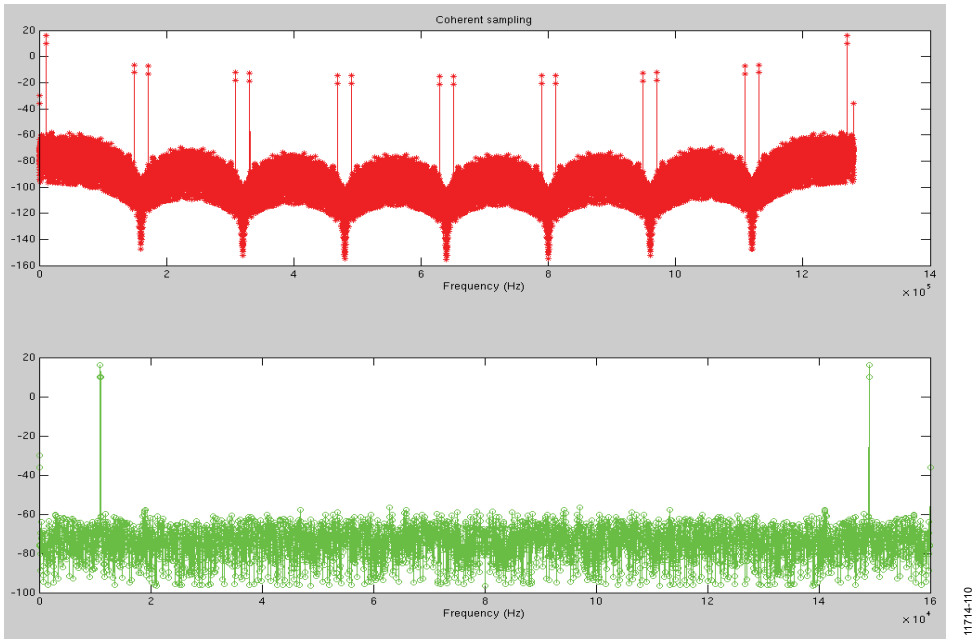


Figure 111. Coherent Sampling

One way to solve this is to skew the DAC and ADC clocks with respect to each other, such that neither of them is an integer multiple of each other. Figure 112 shows a noncoherent alternative to the setup in Figure 110. Here the ADC sampling frequency has been modified to be 142.2 kHz (160 kHz × 8/9), and the DAC update rate remains 160 kHz. Rather than being aliased in the same frequency bin as the fundamental, the images sit in their own bins.

Note that the ADC sampling frequency and DAC update rate used in the previous examples (see Figure 110) are for illustration purposes only.

Modifying the DAC update rate is preferable because it does not require the filtering changes that modifying the ADC sampling frequency entails.

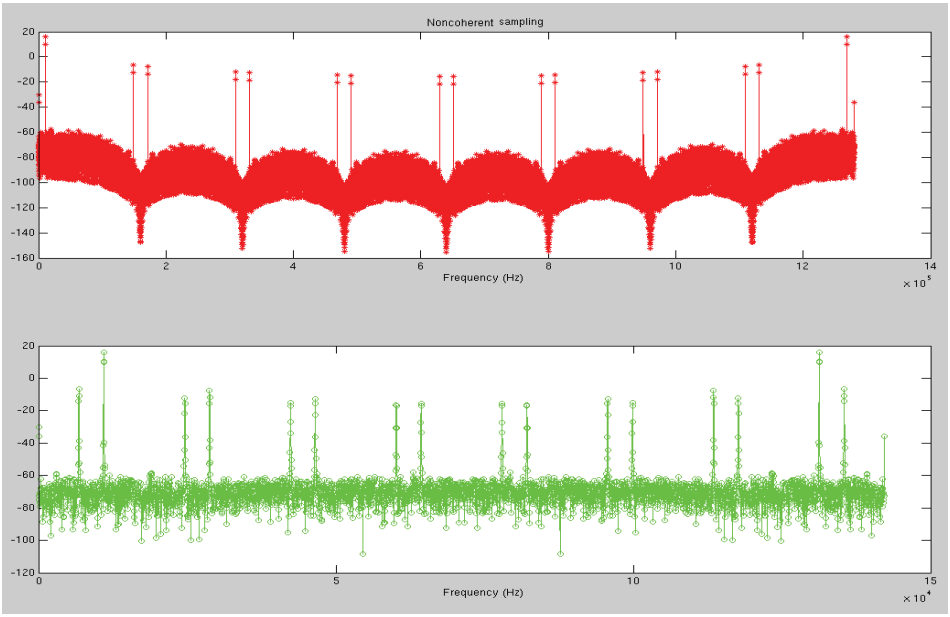


Figure 112. Noncoherent Sampling

By default, the DAC update rate is the ACLK divided by 50; 16 MHz/50 = 320 kHz. The AFE allows programmability of the DAC update divide ratio through the DAC_UPDATE_RATE bits in the AFE_DAC_CFG register. The programmable DAC update rate can be between 280.7 kHz (divide ratio = 57) and 380.9 kHz (divide ratio = 42).

Note that the DAC_UPDATE_RATE bit applies only to sinusoid generation. For anything else (for example, trapezoid), the update rate is fixed at 50 (320 kHz).

When changing the DAC update rate, the DDS engine parameters (frequency control word and phase offset) can be tweaked such that the output sinusoid is the same as with the previous DAC update rate.

The ADC sampling frequency is always fixed at 160 kHz.

The DFT process is very immune to noise because it is highly selective. To allow the user to put some fail-safe measures around the quality of the signal being measured in the DFT addition checking of the ADC samples, set the ADC_MAX_FAIL_IEN or ADC_MINFAIL_IEN bit in the AFE_ANALOG_CAPTURE_IEN register.

50 Hz/60 Hz Supply Rejection Filter

This block implements a low-pass supply rejection filter for output data rates of 900 SPS. The requirement is to reject 50 Hz/60 Hz tones in the dc amperometric phase.

The filter block consists of two cascaded sinc2 filters. The first filter (sinc2hf) decimates the input data (ADC data sampled at 160 kHz) down to ~900 Hz. The second filter (sinc2lf) adds notches at 50 Hz and 60 Hz for supply rejection. Typically, the filtered data (sinc2lf) is supplied to the Cortex-M3, but the option exists to read back the unfiltered data (sinc2hf output).

The filter settling time is 1/50 + 1/60 = 36.667 ms.

The block diagram is shown in Figure 113.

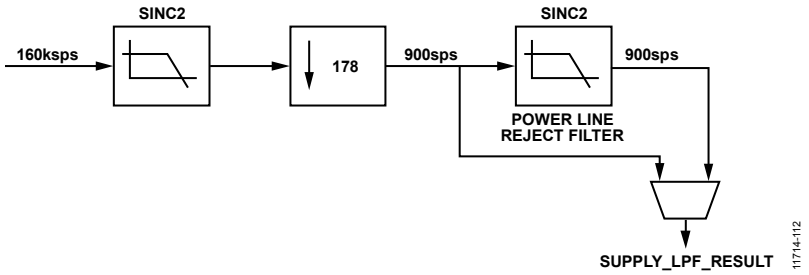


Figure 113. Supply Rejection Filter Block Diagram

The sinc2hf filter decimates the 160 kHz input data by 178 for a target output data rate of 900 Hz. The reason for choosing 900 Hz is to allow the optimal placement of the 50 Hz and 60 Hz notches. Decimation by 178 brings the data rate to 898.87 Hz, which is as close as possible to 900 Hz through integer division. The sinc2hf settling time is 2 × 1/(160000/178) = 2.22 ms.

Selecting a 1 for the BYPASS_SUPPLY_LPF bit (Bit 0) in the AFE_SUPPLY_LPF_CFG register bypasses the 50 Hz/60 Hz rejection filter and enables the sinc2hf filter only. Selecting a 0 enables both the sinc2lf and sinc2hf filters.

Figure 114 and Figure 115 show the frequency response of the sinc2hf filter.

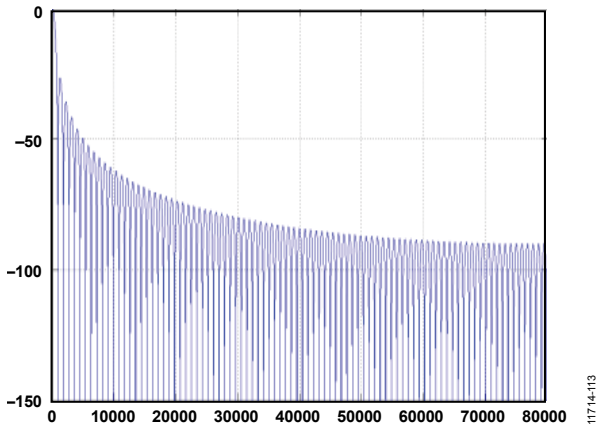


Figure 114. Sinc2hf Frequency Response

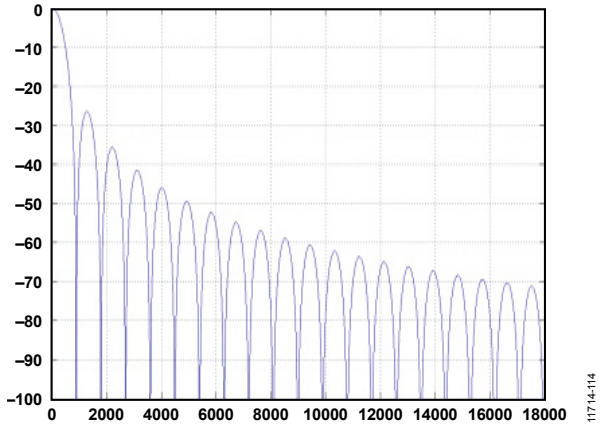


Figure 115. Sinc2hf Frequency Response Detail

The second sinc filter (sinc2lf) adds notches at 50 Hz and 60 Hz. There is no decimation, and the output data rate is the same as the input data rate (900 Hz). The impulse response of the filter is shown in Figure 116, and the frequency response is shown in Figure 117.

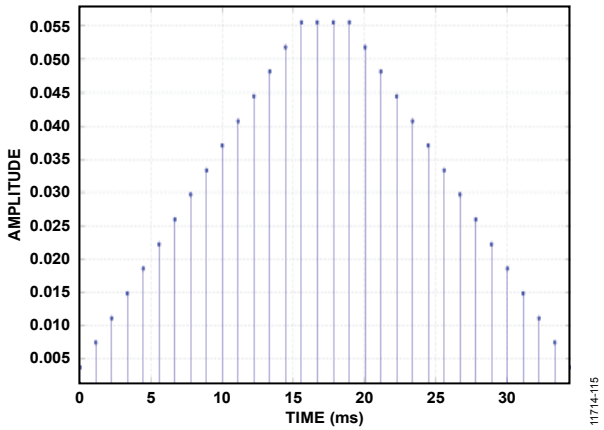


Figure 116. Sinc2lf Impulse Response

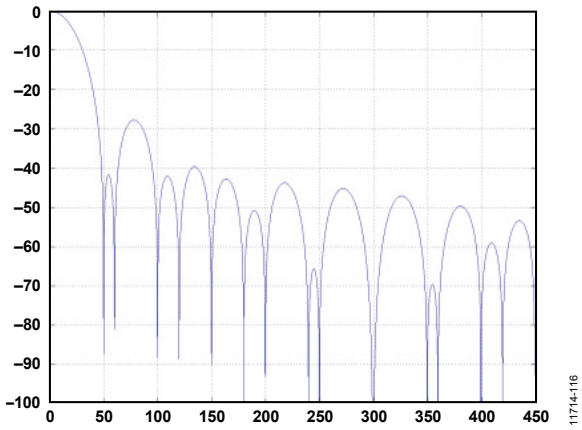


Figure 117. Sinc2lf Frequency Response

The sinc2lf settling time is 31 × 1/(160000/178) = 34.48 ms.

The frequency response of the composite filter is shown in Figure 118 and Figure 119.

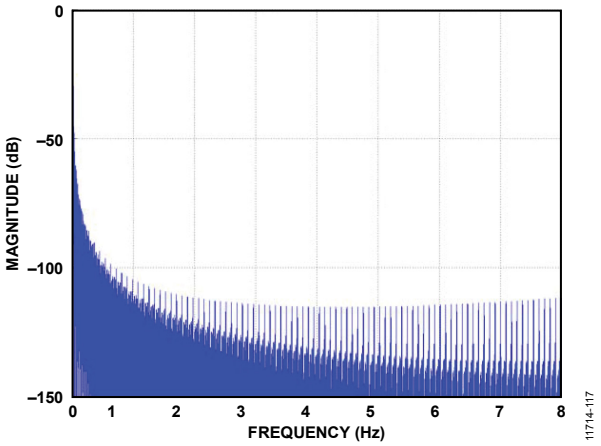


Figure 118. Composite Filter Frequency Response

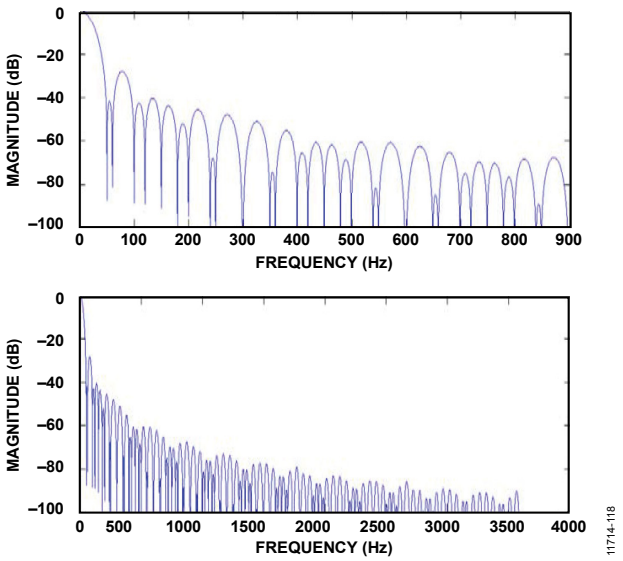


Figure 119. Composite Filter Frequency Response Details

The filter is enabled by writing a 1 to the SUPPLY_LPF_EN bit in the AFE_CFG register. After the filter is enabled, the block waits for a new sample from the ADC and processes it. When a new output is available, the flag AFE_SUPPLY_LPF_RESULT_STATUS is activated, and the respective interrupt is triggered. The filter continues to process data as long as SUPPLY_LPF_EN = 1.

Output data from this filter block can be read directly from the AFE_SUPPLY_LPF_RESULT register or can be written into the data FIFO (DATA_FIFO_SOURCE_SEL = 11) to be read through DMA transfers.

When enabled (SUPPLY_LPF_EN changes from 0 to 1), the internal state of the filter is reset.

ADC Decimation Filter

The ADC sampling rate is 800 kHz, which is then decimated down by 5 to achieve the 160 kHz data rate for the DSP hardware accelerators. The decimation is done using a sinc3 filter, as shown in Figure 120.

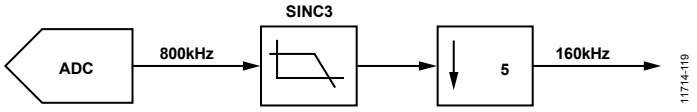


Figure 120. ADC Decimation Filter

The ADC decimation filter has notches at integer multiples of 160 kHz, except for 800 kHz. Filtering of interferes at integer multiples of 800 kHz is done by the analog antialiasing filter.

Data FIFO

The data FIFO provides a buffer for the output of the analog/DSP blocks before it is read by the Cortex-M3. It has a depth of eight words of 16 bits each. It is always unidirectional: a selectable source in the AFE block writes data, and the Cortex-M3 reads data through the AFE_DATA_FIFO_READ register.

The FIFO is synchronous: ACLK is used for writing, and HCLK_BUS is used for reading. Therefore, there is no need for synchronization.

A number of flags are associated with the data FIFO, including the following:

- Empty (DATA_FIFO_EMPTY_STATUS): 1 if there are no words in the FIFO; 0 otherwise.
- Full (DATA_FIFO_FULL_STATUS): 1 if there are eight words in the FIFO; 0 otherwise.
- Overflow (DATA_FIFO_OVF_STATUS): set to 1 for 1 clock period if a data source tries to write a word to the FIFO when the FIFO is full; 0 otherwise.
- Underflow (DATA_FIFO_UDF_STATUS): set to 1 for 1 clock period if the Cortex-M3 tries to read from an empty FIFO; 0 otherwise.
- Threshold (DATA_FIFO_THR_STATUS): 1 if the number of words in the FIFO is greater than the threshold set in the DATA_FIFO_THR_VAL bit (Register AFE_FIFO_CFG); 0 otherwise. This flag is also 0 if the FIFO is full.

The flags are user readable through the AFE_INT_STATUS. Each of the flags has a maskable interrupt associated with it.

The overflow (DATA_FIFO_OVF_STATUS) and underflow (DATA_FIFO_UDF_STATUS) flags only activate for 1 clock period. However, the state of their corresponding interrupt bits (DATA_FIFO_OVF/DATA_FIFO_UDF) indicates whether an overflow or underflow has occurred at any point during the FIFO operation.

The DATA_FIFO_EN bit (Register AFE_FIFO_CFG) enables the FIFO when a 1 is written to it. If DATA_FIFO_EN = 0, the FIFO is reset by resetting the read and write pointers back to their default values (0 for both). The FIFO stays in reset mode until a 1 is written to DATA_FIFO_EN.

There is a DMA channel associated with the FIFO. To enable it, a 1 must be written to the DATA_FIFO_DMA_REQ_EN bit (Register AFE_FIFO_CFG). The DMA channel is disabled by default due to DATA_FIFO_EN being 0 even though DATA_FIFO_DMA_REQ_EN is 1. When the FIFO is enabled (DATA_FIFO_EN = 1) and DATA_FIFO_DMA_REQ_EN = 1, the FIFO issues a DMA request any time it is not empty. The DMA request is deasserted when the FIFO is empty and is asserted again when a new word is written to the FIFO (FIFO is not empty any more).

The number of words in the FIFO can be inspected at any time by reading the DATA_FIFO_WORD_COUNT bits of the AFE_STATUS register.

The data pushed into the FIFO is selectable:

- Raw ADC output
- DFT output
- Supply rejection filter output

The data source is programmed into the DATA_FIFO_SOURCE_SEL bits in the AFE_FIFO_CFG register.

Reading from the data FIFO when empty returns 0x00000000. In addition, the underflow flag (the DATA_FIFO_UDF bit of the AFE_DATA_FIFO_INT register) is asserted.

The block diagram of the data sources for the FIFO is shown in Figure 121. The data rates are for illustrative purposes only.

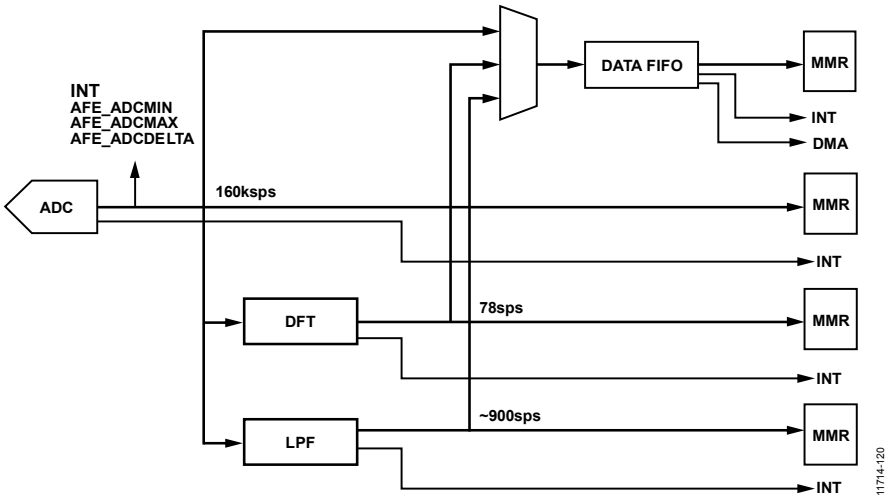


Figure 121. Data FIFO Data Sources

Programmable Value Limits

The raw ADC samples can be set to check that the maximum and minimum ADC samples do not exceed a programmable value defined in the AFE_ADCMIN and AFE_ADCMAX registers. These checks are completed after the gain and offset correction stage. If the values are exceed, the ADC_CHK_FAIL_IEN bit is set if enabled.

The delta between adjacent ADC codes can also be measured. If the delta exceeds the value defined in the AFE_ADCDELTA, the ADC_CHK_FAIL_IEN bit is set if enabled.

ADC Results (AFE_ADC_RESULT)

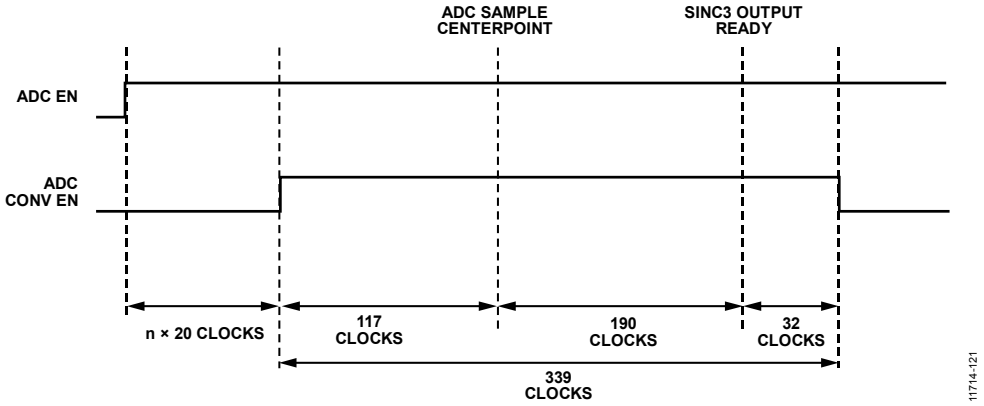


Figure 122. ADC Enable Timing

Note that the defined delay between ADC_EN and ADC_CONV_EN is 100 µs to allow the ADC to settle. This example does not take the extra delay of 100 µs when changing the correct ADC Mux (see the No Factory Calibration section for more information).

It is assumed that between enabling the ADC (ADC_EN = 1) and starting ADC conversions (ADC_CONV_EN = 1) there is a wait time multiple of 20. This guarantees cycle accurate timing of the ADC sample center point. Otherwise, the ADC sample center point varies slightly with respect to ADC_CONV_EN, and the time ADC_CONV_EN stays high is a minimum of 352 clock cycles (equivalent to a wait timer value of 350).

The time between sinc3 output ready and ADC_CONV_EN = 0 (32 clock cycles) is used for output scaling and gain/offset correction, done using serial arithmetic.

If more samples are needed and ADC_CONV_EN remains set to 1, there is a valid ADC sample every 100 clock cycles (equivalent to a 160 kHz sampling frequency).

A sequence that can be used to read a single ADC sample follows:

```
// Enable ADC (ADC_EN = 1)
0x80000080
// Wait 1598 clock cycles (1600 clock cycles actual wait, with 2 extra from executing the
commands themselves)
0x0000063E
// Enable ADC conversion (ADC_CONV_EN = 1)
0x80000180
// Wait 337 clock cycles (339 clock cycles actual wait)
0x00000151
// Disable ADC and ADC_CONV_EN
0x80000000
// Disable sequencer
0x82000000
```

Figure 123 shows the relationship between sampled data points and ADC outputs when there are multiple conversions. ADC Sample 1 is an estimation of the ADC input at Data Point 1, and so on.

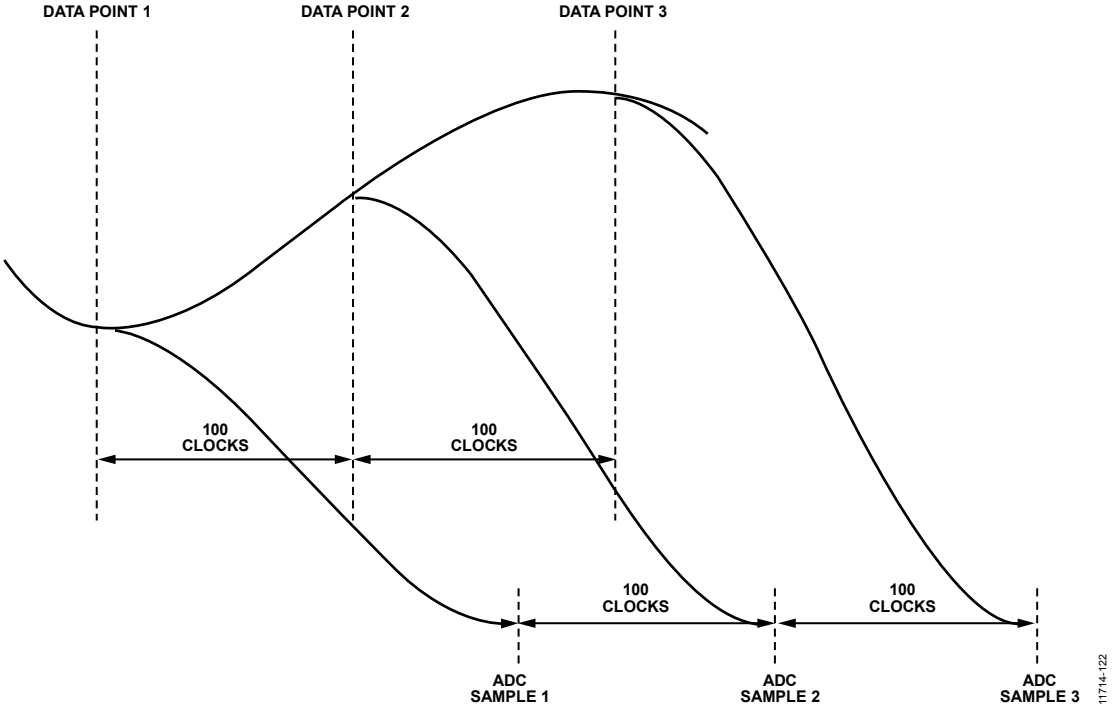


Figure 123. ADC Sampling Data Points vs. Output Data Points

DFT Results (AFE_DFT_RESULT_REAL, AFE_DFT_RESULT_IMAG)

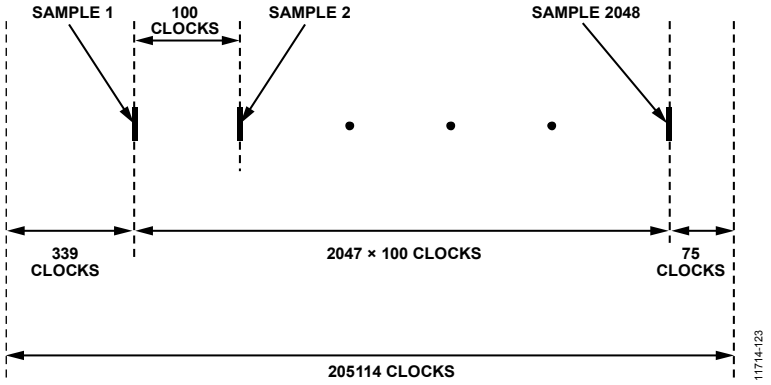


Figure 124. DFT Timing

Discrete Fourier transform timing is shown in Figure 124. The timing of Sample 1 is the timing of the ADC result, as described in the ADC Results (AFE_ADC_RESULT) section. The extra latency of 75 clocks is used for Hann window and Fourier transform calculations.

This timing assumes a 1 is written to DFT_EN at the same time as a 1 is written to ADC_CONV_EN.

A sequence that can be used to produce a valid DFT output follows:

```
// Enable ADC (ADC_EN = 1)
0x80000080
// Wait 1598 clock cycles (1600 clock cycles actual wait, with 2 extra from executing the
commands themselves)
0x00000063E
// Enable ADC conversion (ADC_CONV_EN = 1) and DFT engine (DFT_EN = 1)
0x80008180
// Wait 205112 clock cycles
0x00032138
// Disable ADC, ADC_CONV_EN, and DFT_EN
0x80000000
// Disable sequencer
0x82000000
```

Supply Rejection Filter Result (AFE_SUPPLY_LPF_RESULT)—Sinc2hf and Sinc2lf

The breakup of the timing through the 50 Hz/60 Hz rejection filter chain is as described in Table 471.

Table 471. Timing Through 50 Hz/60 Hz Filter—Sinc2hf and Sinc2lf

Timing Step	Number of Clocks[1]	Details
ADC	339 clocks	
Sinc2hf	2 × 178 × 100 clocks	Decimate by 178 sinc2 filter. The first settled output is the second output.
Sinc2lf	31 × 178 × 100 clocks	50 Hz/60 Hz rejection filter. Uses 32 input samples (sinc2hf outputs) to produce a valid result.
Extra Latency	21 clocks	Used for passing the data between stages and for final result scaling.
Total	587,760 clocks (36.735 ms)	

[1] Refers to the 16 MHz system clock.

A sequence that can be used to produce a valid filtered output follows:

```
// Enable ADC (ADC_EN = 1)
0x80000080
// Wait 1598 clock cycles (1600 clock cycles actual wait, with 2 extra from executing the
commands themselves)
0x00000063E
// Enable ADC conversion (ADC_CONV_EN = 1) and supply rejection filter (SUPPLY_LPF_EN = 1)
0x80010180
// Wait 587,758 clock cycles (587,760 clock cycles actual wait)
0x0008F7EE
// Disable ADC, ADC_CONV_EN, and SUPPLY_LPF_EN
0x80000000
// Disable sequencer
0x82000000
```

Supply Rejection Filter Result (AFE_SUPPLY_LPF_RESULT)—sinc2hf Only

The breakup of the timing through the filter chain when sinc2lf is bypassed (BYPASS_SUPPLY_LPF = 1) is as described in Table 472.

Table 472. Timing Through 50 Hz/60 Hz Filter—Sinc2hf Only

Timing Step	Number of Clocks[1]	Details
ADC	339 clocks	
Sinc2hf	2 × 178 × 100 clocks	Decimate by 178 sinc2 filter. The first settled output is the second output.
Extra Latency	15 clocks	Used for passing the data between stages and for final result scaling.
Total	35,954 (2.247 ms)	

[1] Refers to the 16 MHz system clock.

Note that in the sequencer examples, some bits are ignored (for example, ALDO_EN and REF_EN). It is assumed that all required blocks are turned on.

Calibration Registers

Several registers are needed for the analog blocks calibration and trim data. These registers are located in the addresses starting at 0x40080100.

It is important to note that the calibration registers are not accessible through the sequencer. Writing to these registers is only possible by using direct writes to the Cortex-M3.

The values in the calibration registers can either result from factory calibration (and be stored in flash) or result from calibration routines performed when the AFE powers up. These registers are protected by a password located in the AFE_CAL_DATA_LOCK register. The registers must be unlocked before writing to them. If a write is attempted to the calibration registers while they are locked, an error is flagged on the AHB bus.

Note that the offset and gain registers for the ADC sensor/temperature/auxiliary channel measurements are not retained when the device is in hibernate mode; these registers must be set as 1 (AFE_ADC_GAIN_X) and 0 (AFE_ADC_OFFSET_X) in the autocalibration routine. All other calibration registers are retained when the device is in hibernate mode.

AFE Trim Registers

The AFE trim values are loaded by the kernel into their respective MMRs upon power-up. To facilitate user safety checks at any time, the locations in flash where the AFE trim values are stored are user readable. Table 473 shows the mapping between the AFE trim registers and the flash address space.

Table 473. User Readable AFE Trim Locations

Register Name	Flash Address	Number of Bytes
AFE_DAC_OFFSET_UNITY	0x607C0	2
AFE_DAC_OFFSET_ATTEN	0x607C2	2
AFE_DAC_GAIN	0x607C4	2
AFE_REF_TRIM0	0x607C6	2
AFE_REF_TRIM1	0x607C8	4
AFE_ALDO_TRIM	0x607CC	1
AFE_DAC_TRIM	0x607CE	2
AFE_EXBUF_TRIM	0x607D0	4
AFE_TEMP_SENS_TRIM	0x607D4	2
AFE_INAMP_TRIM	0x607D6	2

Sequencer Startup

To start the sequencer, the Cortex-M3 first must program the DMA controller to read SRAM or flash locations and write them to the command FIFO (Register AFE_CMD_FIFO_WRITE). Then the command FIFO is enabled by writing to the AFE_FIFO_CFG register (CMD_FIFO_EN = 1 and CMD_FIFO_DMA_REQ_EN = 1). This causes the FIFO controller to check the FIFO status, detect the FIFO is empty, and assert the DMA request.

At a moment chosen by the user (for example, when the CMD_FIFO_FULL flag is asserted), a 1 is written to SEQ_EN in the AFE_SEQ_CFG register. This starts the sequencer that reads the commands from the FIFO and executes them sequentially. Each time the command FIFO is not full, the DMA request is asserted, and a new command is transferred by the DMA controller.

Power-Down Considerations

The recommended power-down sequence for the AFE when the device is going into hibernate mode is shown in Table 474.

Table 474. AFE Hibernate Mode Sequence

Action	Register	Address	Value	Notes
Disable Sequencer	AFE_SEQ_CFG	0x40080004	0x00000000	Disable sequencer.
Disable Analog Blocks and Hardware Accelerators	AFE_CFG	0x40080000	0x00000003	This value leaves all hardware accelerators and analog blocks disabled.
Disable DMA				
	AFE_FIFO_CFG	0x40080008	0x00000000	This resets the FIFO. If data is useful, it must be read before the device goes into hibernate mode.
	DMAENCLR	0x4001002C	Value depends on the DMAs enabled	Disable AFE DMA channels.
Disable Interrupts				
	AFE_ANALOG_CAPTURE_IEN	0x4008008C	0x00000000	
	AFE_ANALOG_GEN_IEN	0x40080090	0x00000000	
	AFE_CMD_FIFO_IEN	0x40080094	0x00000000	
	AFE_DATA_FIFO_IEN	0x40080098	0x00000000	
Disable ACLK	CLKCON5	0x40028014	Value depends on the DMAs enabled	ACLKOFF = 1. Not strictly needed, but safer and saves power.

AFE MEMORY MAPPED REGISTERS

AFE Register Map

Table 475. AFE Register Summary

Address	Name	Description	Reset	RW
0x40080000	AFE_CFG	AFE configuration	0x00000000	RW
0x40080004	AFE_SEQ_CFG	Sequencer configuration	0x00000002	RW
0x40080008	AFE_FIFO_CFG	FIFOs configuration	0x00001010	RW
0x4008000C	AFE_SW_CFG	Switch matrix configuration	0x00000000	RW
0x40080010	AFE_DAC_CFG	DAC configuration	0x00000F00	RW
0x40080014	AFE_WG_CFG	Waveform generator configuration	0x00000030	RW
0x40080018	AFE_WG_DCLEVEL_1	Waveform generator—Trapezoid DC Level 1	0x00000000	RW
0x4008001C	AFE_WG_DCLEVEL_2	Waveform generator—Trapezoid DC Level 2	0x00000000	RW
0x40080020	AFE_WG_DELAY_1	Waveform generator—Trapezoid Delay 1 time	0x00000000	RW
0x40080024	AFE_WG_SLOPE_1	Waveform generator—Trapezoid Slope 1 time	0x00000000	RW
0x40080028	AFE_WG_DELAY_2	Waveform generator—Trapezoid Delay 2 time	0x00000000	RW
0x4008002C	AFE_WG_SLOPE_2	Waveform generator—Trapezoid Slope 2 time	0x00000000	RW
0x40080030	AFE_WG_FCW	Waveform generator—sinusoid frequency control word	0x00000000	RW
0x40080034	AFE_WG_PHASE	Waveform generator—sinusoid phase offset	0x00000000	RW
0x40080038	AFE_WG_OFFSET	Waveform generator—sinusoid offset	0x00000000	RW
0x4008003C	AFE_WG_AMPLITUDE	Waveform generator—sinusoid amplitude	0x00000000	RW
0x40080040	AFE_ADC_CFG	ADC configuration	0x00000000	RW
0x40080044	AFE_SUPPLY_LPF_CFG	Supply rejection filter configuration	0x00000000	RW
0x40080048	AFE_SW_FULL_CFG_MSB	Switch matrix full configuration (MSB)	0x00000000	RW
0x4008004C	AFE_SW_FULL_CFG_LSB	Switch matrix full configuration (LSB)	0x00000000	RW
0x40080054	AFE_WG_DAC_CODE	Waveform generator—DAC code	0x00000800	RW
0x40080058	AFE_STATUS	AFE status	0x00001010	R
0x40080060	AFE_SEQ_CRC	Sequencer CRC value	0x00000001	R
0x40080064	AFE_SEQ_COUNT	Sequencer command count	0x00000000	RW
0x40080068	AFE_SEQ_TIMEOUT	Sequencer timeout counter	0x00000000	R
0x4008006C	AFE_DATA_FIFO_READ	Data FIFO read	0x00000000	R
0x40080070	AFE_CMD_FIFO_WRITE	Command FIFO write	0x00000000	W
0x40080074	AFE_ADC_RESULT	ADC raw result	0x00000000	RW
0x40080078	AFE_DFT_RESULT_REAL	DFT result, real part	0x00000000	RW
0x4008007C	AFE_DFT_RESULT_IMAG	DFT result, imaginary part	0x00000000	RW
0x40080080	AFE_SUPPLY_LPF_RESULT	Supply rejection filter result	0x00000000	RW
0x40080084	AFE_TEMP_SENSOR_RESULT	Temperature sensor result	0x00000000	RW
0x4008008C	AFE_ANALOG_CAPTURE_IEN	Analog capture interrupt enable	0x00000000	RW
0x40080090	AFE_ANALOG_GEN_IEN	Analog generation interrupt enable	0x00000000	RW
0x40080094	AFE_CMD_FIFO_IEN	Command FIFO interrupt enable	0x00000000	RW
0x40080098	AFE_DATA_FIFO_IEN	Data FIFO interrupt enable	0x00000000	RW
0x400800A0	AFE_ANALOG_CAPTURE_INT	Analog capture interrupt	0x00000000	RW1C
0x400800A4	AFE_ANALOG_GEN_INT	Analog generation interrupt	0x00000000	RW1C
0x400800A8	AFE_CMD_FIFO_INT	Command FIFO interrupt	0x00000010	RW1C
0x400800AC	AFE_DATA_FIFO_INT	Data FIFO interrupt	0x00000002	RW1C
0x400800B0	AFE_SW_STATUS_MSB	Switch matrix status (MSB)	0x00010000	R
0x400800B4	AFE_SW_STATUS_LSB	Switch matrix status (LSB)	0x00010000	R
0x400800B8	AFE_ADCMIN	ADC minimum value check	0x00000000	RW
0x400800BC	AFE_ADCMAX	ADC maximum value check	0x00000000	RW
0x400800C0	AFE_ADCDELTA	ADC delta check	0x00000000	RW
0x40080100	AFE_CAL_DATA_LOCK	Calibration data lock	0x00000000	RW
0x40080104	AFE_ADC_GAIN_TIA	ADC gain (TIA measurement)	0x00004000	RW
0x40080108	AFE_ADC_OFFSET_TIA	ADC offset (TIA measurement)	0x00000000	RW
0x4008010C	AFE_ADC_GAIN_TEMP_SENS	ADC gain (temperature sensor measurement)	0x00004000	RW

Address	Name	Description	Reset	RW
0x40080110	AFE_ADC_OFFSET_TEMP_SENS	ADC offset (temperature sensor measurement)	0x00000000	RW
0x40080118	AFE_ADC_GAIN_AUX	ADC gain (auxiliary channel measurement)	0x00004000	RW
0x4008011C	AFE_ADC_OFFSET_AUX	ADC offset (auxiliary channel measurement)	0x00000000	RW
0x40080120	AFE_DAC_OFFSET_UNITY	DAC offset with attenuator disabled	0x00000000	RW
0x40080124	AFE_DAC_OFFSET_ATTEN	DAC offset with attenuator enabled	0x00000000	RW
0x40080128	AFE_DAC_GAIN	DAC gain	0x00000800	RW
0x4008012C	AFE_REF_TRIM0	Precision reference trim 0	0x00000000	RW
0x40080130	AFE_REF_TRIM1	Precision reference trim 1	0x00000000	RW
0x40080134	AFE_ALDO_TRIM	Analog LDO trim	0x00000000	RW
0x40080138	AFE_DAC_TRIM	DAC trim	0x00000000	RW
0x4008013C	AFE_INAMP_TRIM	INAMP trim	0x00000000	RW
0x40080140	AFE_EXBUF_TRIM	Excitation buffer trim	0x00000000	RW
0x40080144	AFE_TEMP_SENS_TRIM	Temperature sensor trim	0x00004000	RW

AFE Configuration Register

Address: 0x40080000, Reset: 0x00000000, Name: AFE_CFG

Table 476. Bit Descriptions for AFE_CFG

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x000	R
19	ALDOILIMIT_EN	Enable AFE analog LDO buffer current limiting. If enabled, it limits the current drawn from the battery while charging the cap on AVDDRX and AVDDTX. 0: analog LDO buffer current limiting disabled. 1: analog LDO buffer current limiting enabled.	0x0	RW
18	VREFBUFILIMIT_EN	Enable VREF buffer current limiting. If enabled, it limits the current drawn from the battery while charging the cap on VREF. 0: VREF buffer current limiting disabled. 1: VREF buffer current limiting enabled.	0x0	RW
17	VBIASBUF_EN	Enable VBIAS buffer. 0: VBIAS buffer disabled. 1: VBIAS buffer enabled.	0x0	RW
16	SUPPLY_LPF_EN	Enable 50 Hz/60 Hz supply rejection filter. 0: supply rejection filter disabled. 1: supply rejection filter enabled.	0x0	RW
15	DFT_EN	Enable the DFT hardware acceleration block. 0: DFT hardware accelerator disabled. 1: DFT hardware accelerator enabled.	0x0	RW
14	WAVEGEN_EN	Enable waveform generator. 0: waveform generator disabled. 1: waveform generator enabled.	0x0	RW
13	TEMP_CONV_EN	Enable temperature reading. If 1, a temperature reading is initiated. When the temperature conversion is complete and the result available in AFE_TEMP_SENSOR_RESULT register, this bit is reset to 0. 0: temperature reading disabled. 1: temperature reading enabled.	0x0	RW
12	TEMP_SENSOR_EN	Enable temperature sensor. If 1, the temperature sensor is powered up but no temperature reading is performed unless TEMP_CONV_EN = 1. 0: temperature sensor disabled. 1: temperature sensor enabled.	0x0	RW
11	TIA_EN	Enable transimpedance amplifier. 0: transimpedance amplifier disabled. 1: transimpedance amplifier enabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
10	INAMP_EN	Enable instrumentation amplifier. 0: instrumentation amplifier disabled. 1: instrumentation amplifier enabled.	0x0	RW
9	BUF_EN	Enable excitation buffer. 0: excitation buffer disabled. 1: excitation buffer enabled.	0x0	RW
8	ADC_CONV_EN	Enable ADC conversions. If 1, the ADC is converting, otherwise the ADC is idle. 0: ADC idle. 1: ADC performing conversions.	0x0	RW
7	ADC_EN	Enable ADC. If 1, the ADC is powered up but no conversion is performed unless the ADC_CONV_EN bit is set. 0: ADC disabled. 1: ADC enabled.	0x0	RW
6	DAC_EN	Enable DAC, reconstruction filter and attenuator. 0: DAC disabled. 1: DAC enabled.	0x0	RW
5	REF_EN	Enable internal reference. 0: reference disabled. 1: reference enabled.	0x0	RW
4	ALDO_EN	Enable analog LDO. 0: analog LDO disabled. 1: analog LDO enabled.	0x0	RW
[3:0]	RESERVED	Reserved.	0x0	R

Sequencer Configuration Register

Address: 0x40080004, Reset: 0x00000002, Name: AFE_SEQ_CFG

Table 477. Bit Descriptions for AFE_SEQ_CFG

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0000	R
[15:8]	SEQ_WRITE_TIMER	Timer for sequencer write commands. It acts as a clock divider affecting the write commands, but not the wait commands. This is useful to reduce the code size when generating arbitrary waveforms.	0x00	RW
[7:5]	RESERVED	Reserved.	0x0	R
4	SEQ_HALT	Debugging feature, provides a way to halt the AFE interface, including the sequencer, DSP hardware accelerators, FIFOs, etc. 0: normal execution. 1: execution halted.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	R
1	SEQ_STOP_ON_FIFO_EMPTY	Controls if the sequencer stops when trying to read when the command FIFO is empty (underflow condition). 1: sequencer stops if command FIFO is empty and sequencer tries to read (underflow condition). 0: sequencer keeps trying to read even if FIFO is empty.	0x1	RW
0	SEQ_EN	Enable sequencer. If 1, the sequencer reads from the command FIFO and executes the commands. The bit is set by the Cortex-M3, and reset by either the Cortex-M3 or the sequencer as the result of command execution. 0: sequencer disabled. 1: sequencer enabled.	0x0	RW

FIFOs Configuration Register

Address: 0x40080008, Reset: 0x00001010, Name: AFE_FIFO_CFG

Table 478. Bit Descriptions for AFE_FIFO_CFG

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x0000	R
[14:13]	DATA_FIFO_SOURCE_SEL	Selects the source for the data FIFO. 0X: ADC 10: DFT 11: supply rejection filter.	0x0	RW
12	DATA_FIFO_DMA_REQ_EN	Enable data FIFO DMA channel. 0: disable DMA requests for data FIFO. 1: enable DMA requests for data FIFO.	0x1	RW
11	DATA_FIFO_EN	Data FIFO enable. 0: FIFO is reset. No data transfers may take place. Sets the read and write pointers to the default values (empty FIFO). The status indicates FIFO is empty. 1: normal operation. FIFO is not reset.	0x0	RW
[10:8]	DATA_FIFO_THR_VAL	Data FIFO threshold. When the number of words in the FIFO is above the threshold value, the threshold flag activates. This can be used to trigger an interrupt or a DMA request.	0x0	RW
[7:5]	RESERVED	Reserved.	0x0	R
4	CMD_FIFO_DMA_REQ_EN	Enable command FIFO DMA channel. 0: disable DMA requests for command FIFO. 1: enable DMA requests for command FIFO.	0x1	RW
3	CMD_FIFO_EN	Command FIFO enable. 0: FIFO is reset. No data transfers may take place. Sets the read and write pointers to the default values (empty FIFO). The status indicates FIFO is empty. 1: normal operation. FIFO is not reset.	0x0	RW
[2:0]	CMD_FIFO_THR_VAL	Command FIFO threshold. When the number of words in the FIFO is below the threshold value, the threshold flag activates. This can be used to trigger an interrupt or a DMA request.	0x0	RW

Switch Matrix Configuration Register

Address: 0x4008000C, Reset: 0x00000000, Name: AFE_SW_CFG

Table 479. Bit Descriptions for AFE_SW_CFG

Bits	Bit Name	Description	Reset	Access
[31:18]	RESERVED	Reserved.	0x0	R
17	SW_SOURCE_SEL	1: load switch matrix control register from the AFE_SW_FULL_CFG_MSB/ AFE_SW_FULL_CFG_LSB registers. 0: load switch matrix control register from AFE_SW_CFG.	0x0	RW
16	IVS_STATE	0: IVS switch open. 1: IVS switch closed.	0x0	RW
[15:12]	TMUX_STATE	0000: all switches open. 0001: T1 closed, rest open. 0010: T2 closed, rest open. 0011: T3 closed, rest open. 0100: T4 closed, rest open. 0101: T5 closed, rest open. 0110: T6 closed, rest open. 0111: T7 closed, rest open. 1000: TR2 closed, rest open. 1001: all switches closed. 1010 to 1111: all switches open.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[11:8]	NMUX_STATE	0000: NL closed, rest open. 0001: N1 closed, rest open. 0010: N2 closed, rest open. 0011: N3 closed, rest open. 0100: N4 closed, rest open. 0101: N5 closed, rest open. 0110: N6 closed, rest open. 0111: N7 closed, rest open. 1000: NR2 closed, rest open. 1001 to 1110: NL closed, rest open. 1111: all switches open.	0x0	RW
[7:4]	PMUX_STATE	0000: PL closed, rest open. 0001: PR1 closed, rest open. 0010: P2 closed, rest open. 0011: P3 closed, rest open. 0100: P4 closed, rest open. 0101: P5 closed, rest open. 0110: P6 closed, rest open. 0111: P7 closed, rest open. 1000: P8 closed, rest open. 1001 to 1110: PL closed, rest open. 1111: all switches open.	0x0	RW
[3:0]	DMUX_STATE	0000: all switches open. 0001: DR1 closed, rest open. 0010: D2 closed, rest open. 0011: D3 closed, rest open. 0100: D4 closed, rest open. 0101: D5 closed, rest open. 0110: D6 closed, rest open. 0111: D7 closed, rest open. 1000: D8 closed, rest open. 1001: all switches closed. 1010-1111: all switches open.	0x0	RW

DAC Configuration Register

Address: 0x40080010, Reset: 0x00000F00, Name: AFE_DAC_CFG

Table 480. Bit Descriptions for AFE_DAC_CFG

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x0	R
[11:8]	DAC_UPDATE_RATE	DAC update rate. 0111: ACLK/57. 0110: ACLK/56. 0101: ACLK/55. 0100: ACLK/54. 0011: ACLK/53. 0010: ACLK/52. 0001: ACLK/51. 0000: ACLK/50. 1111: ACLK/49. 1110: ACLK/48. 1101: ACLK/47. 1100: ACLK/46. 1011: ACLK/45. 1010: ACLK/44. 1001: ACLK/43. 1000: ACLK/42.	0xF	RW
[7:1]	RESERVED	Reserved.	0x00	R
0	DAC_ATTEN_EN	Enable the attenuator at the output of the DAC. 0: DAC attenuator disabled. 1: DAC attenuator enabled.	0x0	RW

Waveform Generator Configuration Register

Address: 0x40080014, Reset: 0x00000030, Name: AFE_WG_CFG

Table 481. Bit Descriptions for AFE_WG_CFG

Bits	Bit Name	Description	Reset	Access
[31:6]	RESERVED	Reserved.	0x0000000	R
5	USE_DAC_GAIN	Use the DAC gain calculated during the Analog Devices factory trim and stored in AFE_DAC_GAIN. 0: bypass DAC gain correction. 1: perform DAC gain correction.	0x1	RW
4	USE_DAC_OFFSET	Use the DAC offset calculated during the calibration routine and stored in AFE_DAC_OFFSET_UNITY/AFE_DAC_OFFSET_ATTEN. 0: bypass DAC offset correction. 1: perform DAC offset correction.	0x1	RW
3	RESERVED	Reserved.	0x0	R
[2:1]	TYPE_SEL	Selects the type of waveform. Possible options are sinusoid, trapezoid, or direct write to the DAC. 0X: direct write to DAC. 10: sinusoid. 11: trapezoid.	0x0	RW
0	WG_TRAP_RESET	Resets the trapezoid waveform generator, causing the output to restart from the beginning of Delay 1 period, with an output corresponding to dc Level 1. The reset takes effect immediately. After the trapezoid generator is reset, the bit value returns to 0.	0x0	RW

Waveform Generator—Trapezoid DC Level 1 Register

Address: 0x40080018, Reset: 0x00000000, Name: AFE_WG_DCLEVEL_1

Table 482. Bit Descriptions for AFE_WG_DCLEVEL_1

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x0	R
[11:0]	TRAP_DC_LEVEL_1	DC Level 1 value for trapezoid waveform generation.	0x0	RW

Waveform Generator—Trapezoid DC Level 2 Register

Address: 0x4008001C, Reset: 0x00000000, Name: AFE_WG_DCLEVEL_2

Table 483. Bit Descriptions for AFE_WG_DCLEVEL_2

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x0	R
[11:0]	TRAP_DC_LEVEL_2	DC Level 2 value for trapezoid waveform generation.	0x0	RW

Waveform Generator—Trapezoid Delay 1 Time Register

Address: 0x40080020, Reset: 0x00000000, Name: AFE_WG_DELAY_1

Table 484. Bit Descriptions for AFE_WG_DELAY_1

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
[19:0]	TRAP_DELAY_1	Delay 1 value for trapezoid waveform generation. Time unit is the DAC update rate.	0x0	RW

Waveform Generator—Trapezoid Slope 1 Time Register

Address: 0x40080024, Reset: 0x00000000, Name: AFE_WG_SLOPE_1

Table 485. Bit Descriptions for AFE_WG_SLOPE_1

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
[19:0]	TRAP_SLOPE_1	Slope 1 value for trapezoid waveform generation. Time unit is the DAC update rate.	0x0	RW

Waveform Generator—Trapezoid Delay 2 Time Register

Address: 0x40080028, Reset: 0x00000000, Name: AFE_WG_DELAY_2

Table 486. Bit Descriptions for AFE_WG_DELAY_2

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
[19:0]	TRAP_DELAY_2	Delay 2 value for trapezoid waveform generation. Time unit is the DAC update rate.	0x0	RW

Waveform Generator—Trapezoid Slope 2 Time Register

Address: 0x4008002C, Reset: 0x00000000, Name: AFE_WG_SLOPE_2

Table 487. Bit Descriptions for AFE_WG_SLOPE_2

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
[19:0]	TRAP_SLOPE_2	Slope 2 value for trapezoid waveform generation. Time unit is the DAC update rate.	0x0	RW

Waveform Generator—Sinusoid Frequency Control Word Register

Address: 0x40080030, Reset: 0x00000000, Name: AFE_WG_FCW

Table 488. Bit Descriptions for AFE_WG_FCW

Bits	Bit Name	Description	Reset	Access
[31:19]	RESERVED	Reserved.	0x0	R
[18:0]	SINE_FCW	Sinusoid generator frequency control word. Selects the output frequency of the sinusoid.	0x0	RW

Waveform Generator—Sinusoid Phase Offset Register

Address: 0x40080034, Reset: 0x00000000, Name: AFE_WG_PHASE

Table 489. Bit Descriptions for AFE_WG_PHASE

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
[19:0]	SINE_PHASE_OFFSET	Sinusoid phase offset. A value of 0x00000 corresponds to a phase offset of 0°.	0x0	RW

Waveform Generator—Sinusoid Offset Register

Address: 0x40080038, Reset: 0x00000000, Name: AFE_WG_OFFSET

Table 490. Bit Descriptions for AFE_WG_OFFSET

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	R
[11:0]	SINE_OFFSET	Sinusoid offset, added to the waveform generator output in sinusoid mode. Signed number represented in twos complement format.	0x000	RW

Waveform Generator—Sinusoid Amplitude Register

Address: 0x4008003C, Reset: 0x00000000, Name: AFE_WG_AMPLITUDE

Table 491. Bit Descriptions for AFE_WG_AMPLITUDE

Bits	Bit Name	Description	Reset	Access
[31:11]	RESERVED	Reserved.	0x000000	R
[10:0]	SINE_AMPLITUDE	Unsigned number. Scales the waveform generator in sinusoid mode. The DAC output voltage is determined by this value and DAC_ATTEN (if DAC_ATTEN_EN = 1).	0x000	RW

ADC Configuration Register

Address: 0x40080040, Reset: 0x00000000, Name: AFE_ADC_CFG

Table 492. Bit Descriptions for AFE_ADC_CFG

Bits	Bit Name	Description	Reset	Access
[31:10]	RESERVED	Reserved.	0x000000	R
[9:8]	GAIN_OFFS_SEL	00: Use AFE_ADC_GAIN_TIA/AFE_ADC_OFFSET_TIA. 01: Use AFE_ADC_GAIN_TEMP_SENS/AFE_ADC_OFFSET_TEMP_SENS. 10: Use AFE_ADC_GAIN_AUX/AFE_ADC_OFFSET_AUX. 11: bypass gain/offset correction.	0x0	RW
[7:6]	RESERVED	Reserved.	0x0	R
5	ANEXCITESW_EN	AN_EXCITE switch control. 0: AN_EXCITE switch open. 1: AN_EXCITE switch closed.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[4:0]	MUX_SEL	Select signals for the ADC input multiplexer. 00010: TIA. 00011: internal temperature sensor. 10000: 1/2 VCCM_ANA to VBIAS. 10001: 1/2 VCCM_DIG to VBIAS. 10010: 1/2 VBACK to VBIAS. 01100: AN_VEXCITE. 01000: AN_A. 01001: AN_B. 01010: AN_C. 01011: AN_D. 01101: 1/2 DVDD to VBIAS. 01111: 1/2 AVDD_RX to VBIAS. 01110: 1/2 AVDD_TX to VBIAS. 11000: calibration (ADC_OFFSET_TIA). 11001: calibration (ADC_GAIN_TIA/ADC_GAIN_TEMP_SENS). 11010: calibration (ADC_OFFSET_TEMP_SENS). 11100: calibration (ADC_OFFSET_AUX). 11101: calibration (ADC_GAIN_AUX). 11110: P/N automatic calibration. 00000: floating input. …: all other combinations reserved.	0x00	RW

Supply Rejection Filter Configuration Register

Address: 0x40080044, Reset: 0x00000000, Name: AFE_SUPPLY_LPF_CFG

Table 493. Bit Descriptions for AFE_SUPPLY_LPF_CFG

Bits	Bit Name	Description	Reset	Access
[31:1]	RESERVED	Reserved.	0x0	R
0	BYPASS_SUPPLY_LPF	Controls the output of the supply rejection filter block. The output frequency is the same in both cases (900 Hz). 0: sinc2lf output written to SUPPLY_LPF_RESULT. 1: sinc2hf output written to SUPPLY_LPF_RESULT.	0x0	RW

Switch Matrix Full Configuration (MSB) Register

Address: 0x40080048, Reset: 0x00000000, Name: AFE_SW_FULL_CFG_MSB

This register allows individual control of the AFE switches. The bit names are the same as the switch names in the AFE diagram.

Table 494. Bit Descriptions for AFE_SW_FULL_CFG_MSB

Bits	Bit Name	Description	Reset	Access
[31:17]	RESERVED	Reserved.	0x0000	R
16	PL	PL switch control. 0: switch open. 1: switch closed.	0x0	RW
15	P8	P8 switch control. 0: switch open. 1: switch closed.	0x0	RW
14	P7	P7 switch control. 0: switch open. 1: switch closed.	0x0	RW
13	P6	P6 switch control. 0: switch open. 1: switch closed.	0x0	RW

Bits	Bit Name	Description	Reset	Access
12	P5	P5 switch control. 0: switch open. 1: switch closed.	0x0	RW
11	P4	P4 switch control. 0: switch open. 1: switch closed.	0x0	RW
10	P3	P3 switch control. 0: switch open. 1: switch closed.	0x0	RW
9	P2	P2 switch control. 0: switch open. 1: switch closed.	0x0	RW
8	PR1	PR1 switch control. 0: switch open. 1: switch closed.	0x0	RW
7	D8	D8 switch control. 0: switch open. 1: switch closed.	0x0	RW
6	D7	D7 switch control. 0: switch open. 1: switch closed.	0x0	RW
5	D6	D6 switch control. 0: switch open. 1: switch closed.	0x0	RW
4	D5	D5 switch control. 0: switch open. 1: switch closed.	0x0	RW
3	D4	D4 switch control. 0: switch open. 1: switch closed.	0x0	RW
2	D3	D3 switch control. 0: switch open. 1: switch closed.	0x0	RW
1	D2	D2 switch control. 0: switch open. 1: switch closed.	0x0	RW
0	DR1	DR1 switch control. 0: switch open. 1: switch closed.	0x0	RW

Switch Matrix Full Configuration (LSB) Register

Address: 0x4008004C, Reset: 0x00000000, Name: AFE_SW_FULL_CFG_LSB

This register allows individual control of the AFE switches. The bit names are the same as the switch names in the AFE diagram.

Table 495. Bit Descriptions for AFE_SW_FULL_CFG_LSB

Bits	Bit Name	Description	Reset	Access
[31:18]	RESERVED	Reserved.	0x0	R
17	IVS	IVS switch control. 0: switch open. 1: switch closed.	0x0	RW
16	NL	NL switch control. 0: switch open. 1: switch closed.	0x0	RW
15	NR2	NR2 switch control. 0: switch open. 1: switch closed.	0x0	RW

Bits	Bit Name	Description	Reset	Access
14	N7	N7 switch control. 0: switch open. 1: switch closed.	0x0	RW
13	N6	N6 switch control. 0: switch open. 1: switch closed.	0x0	RW
12	N5	N5 switch control. 0: switch open. 1: switch closed.	0x0	RW
11	N4	N4 switch control. 0: switch open. 1: switch closed.	0x0	RW
10	N3	N3 switch control. 0: switch open. 1: switch closed.	0x0	RW
9	N2	N2 switch control. 0: switch open. 1: switch closed.	0x0	RW
8	N1	N1 switch control. 0: switch open. 1: switch closed.	0x0	RW
7	TR2	TR2 switch control. 0: switch open. 1: switch closed.	0x0	RW
6	T7	T7 switch control. 0: switch open. 1: switch closed.	0x0	RW
5	T6	T6 switch control. 0: switch open. 1: switch closed.	0x0	RW
4	T5	T5 switch control. 0: switch open. 1: switch closed.	0x0	RW
3	T4	T4 switch control. 0: switch open. 1: switch closed.	0x0	RW
2	T3	T3 switch control. 0: switch open. 1: switch closed.	0x0	RW
1	T2	T2 switch control. 0: switch open. 1: switch closed.	0x0	RW
0	T1	T1 switch control. 0: switch open. 1: switch closed.	0x0	RW

Waveform Generator—DAC Code Register

Address: 0x40080054, Reset: 0x00000800, Name: AFE_WG_DAC_CODE

Table 496. Bit Descriptions for AFE_WG_DAC_CODE

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x0	R
[11:0]	DAC_CODE	DAC code, written directly to DAC. Minimum code is 0x000 and maximum code is 0xFFF. Midscale (0x800) corresponds to an output voltage of 0 V.	0x800	RW

AFE Status Register

Address: 0x40080058, Reset: 0x00001010, Name: AFE_STATUS

Table 497. Bit Descriptions for AFE_STATUS

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0000	R
15	DATA_FIFO_UDF_STATUS	Data FIFO underflow flag. 0: data FIFO underflow condition not met. 1: data FIFO underflow condition met.	0x0	R
14	DATA_FIFO_OVF_STATUS	Data FIFO overflow flag. 0: data FIFO overflow condition not met. 1: data FIFO overflow condition met.	0x0	R
13	DATA_FIFO_THR_STATUS	Data FIFO threshold flag. 0: data FIFO word count less than threshold. 1: data FIFO word count greater than or equal to threshold.	0x0	R
12	DATA_FIFO_EMPTY_STATUS	Data FIFO empty flag. 0: data FIFO not empty. 1: data FIFO empty.	0x1	R
11	DATA_FIFO_FULL_STATUS	Data FIFO full flag. 0: data FIFO not full. 1: data FIFO full.	0x0	R
[10:8]	DATA_FIFO_WORD_COUNT	Current number of words in the data FIFO.	0x0	R
7	CMD_FIFO_UDF_STATUS	Command FIFO underflow flag. 0: command FIFO underflow condition not met. 1: command FIFO underflow condition met.	0x0	R
6	CMD_FIFO_OVF_STATUS	Command FIFO overflow flag. 0: command FIFO overflow condition not met. 1: command FIFO overflow condition met.	0x0	R
5	CMD_FIFO_THR_STATUS	Command FIFO threshold flag. 0: command FIFO word count less than threshold. 1: command FIFO word count greater than or equal to threshold.	0x0	R
4	CMD_FIFO_EMPTY_STATUS	Command FIFO empty flag. 0: command FIFO not empty. 1: command FIFO empty.	0x1	R
3	CMD_FIFO_FULL_STATUS	Command FIFO full flag. 0: command FIFO not full. 1: command FIFO full.	0x0	R
[2:0]	CMD_FIFO_WORD_COUNT	Current number of words in the command FIFO.	0x0	R

Sequencer CRC Value Register

Address: 0x40080060, Reset: 0x00000001, Name: AFE_SEQ_CRC

Checksum value calculated from all the commands executed by the sequencer.

Table 498. Bit Descriptions for AFE_SEQ_CRC

Bits	Bit Name	Description	Reset	Access
[31:8]	RESERVED	Reserved.	0x0	R
[7:0]	SEQ_CRC	Sequencer command CRC value. The algorithm used is CRC-8.	0x01	R

Sequencer Command Count Register

Address: 0x40080064, Reset: 0x00000000, Name: AFE_SEQ_COUNT

Command count, incremented by 1 each time the sequencer executes a command.

Table 499. Bit Descriptions for AFE_SEQ_COUNT

Bits	Bit Name	Description	Reset	Access
[15:0]	SEQ_COUNT	Sequencer command count.	0x0	R

Sequencer Timeout Counter Register

Address: 0x40080068, Reset: 0x00000000, Name: AFE_SEQ_TIMEOUT

Table 500. Bit Descriptions for AFE_SEQ_TIMEOUT

Bits	Bit Name	Description	Reset	Access
[29:0]	SEQ_TIMEOUT	Current value of the sequencer Timeout counter.	0x0	R

Data FIFO Read Register

Address: 0x4008006C, Reset: 0x00000000, Name: AFE_DATA_FIFO_READ

Table 501. Bit Descriptions for AFE_DATA_FIFO_READ

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	DATA_FIFO_OUT	Data FIFO read. If data FIFO is empty a read of this register returns 0x00000000.	0x0	R

Command FIFO Write Register

Address: 0x40080070, Reset: 0x00000000, Name: AFE_CMD_FIFO_WRITE

Table 502. Bit Descriptions for AFE_CMD_FIFO_WRITE

Bits	Bit Name	Description	Reset	Access
[31:0]	CMD_FIFO_IN	Command FIFO write. If the command FIFO is written while full, the write is ignored and all current commands are not affected.	0x0	W

ADC Raw Result Register

Address: 0x40080074, Reset: 0x00000000, Name: AFE_ADC_RESULT

Table 503. Bit Descriptions for AFE_ADC_RESULT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED		0x0	R
[15:0]	ADC_RESULT	ADC raw result, 16-bit unsigned number.	0x0	RW

DFT Result, Real Part Register

Address: 0x40080078, Reset: 0x00000000, Name: AFE_DFT_RESULT_REAL

Table 504. Bit Descriptions for AFE_DFT_RESULT_REAL

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	DFT_RESULT_REAL	DFT result, real part, represented in twos complement.	0x0	RW

DFT Result, Imaginary Part Register

Address: 0x4008007C, Reset: 0x00000000, Name: AFE_DFT_RESULT_IMAG

Table 505. Bit Descriptions for AFE_DFT_RESULT_IMAG

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	DFT_RESULT_IMAG	DFT result, imaginary part, represented in twos complement.	0x0	RW

Supply Rejection Filter Result Register

Address: 0x40080080, Reset: 0x00000000, Name: AFE_SUPPLY_LPF_RESULT

Table 506. Bit Descriptions for AFE_SUPPLY_LPF_RESULT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	SUPPLY_LPF_RESULT	Supply rejection filter result. When new data is available, the SUPPLY_LPF_RESULT_READY interrupt is triggered.	0x0	RW

Temperature Sensor Result Register

Address: 0x40080084, Reset: 0x00000000, Name: AFE_TEMP_SENSOR_RESULT

Table 507. Bit Descriptions for AFE_TEMP_SENSOR_RESULT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	TEMPERATURE	Temperature measurement result.	0x0	RW

Analog Capture Interrupt Enable Register

Address: 0x4008008C, Reset: 0x00000000, Name: AFE_ANALOG_CAPTURE_IEN

Table 508. Bit Descriptions for AFE_ANALOG_CAPTURE_IEN

Bits	Bit Name	Description	Reset	Access
[31:7]	RESERVED	Reserved.	0x0000000	R
6	ADC_DELTA_FAIL_IEN	ADC delta value check fail interrupt enable. This bit enables ADC delta value checking (ADC_DELTA_FAIL) to generate an interrupt to the core. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
5	ADC_MAX_FAIL_IEN	ADC maximum value check fail interrupt enable. This bit enables ADC maximum value checking (ADC_MAX_FAIL) to generate an interrupt to the core. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
4	ADC_MIN_FAIL_IEN	ADC minimum value check fail interrupt enable. This bit enables ADC minimum value checking (ADC_MIN_FAIL) to generate an interrupt to the core. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
3	TEMP_RESULT_READY_IEN	Temperature measurement result ready interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
2	SUPPLY_LPF_RESULT_READY_IEN	Supply rejection filter result ready interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
1	DFT_RESULT_READY_IEN	DFT result ready interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
0	ADC_RESULT_READY_IEN	ADC result ready interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Analog Generation Interrupt Enable Register

Address: 0x40080090, Reset: 0x00000000, Name: AFE_ANALOG_GEN_IEN

Table 509. Bit Descriptions for AFE_ANALOG_GEN_IEN

Bits	Bit Name	Description	Reset	Access
[31:4]	RESERVED	Reserved.	0x0000000	R
3	CUSTOM_INT_IEN	Custom interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
2	BREAK_SEQUENCE_ORG_IEN	Break sequence interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
1	HARDWARE_SETUP_DONE_IEN	Hardware setup interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
0	DELAY_COMMAND_DONE_IEN	Delay command interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Command FIFO Interrupt Enable Register

Address: 0x40080094, Reset: 0x00000000, Name: AFE_CMD_FIFO_IEN

Table 510. Bit Descriptions for AFE_CMD_FIFO_IEN

Bits	Bit Name	Description	Reset	Access
[31:8]	RESERVED	Reserved.	0x000000	R
7	CMD_FIFO_UDF_IEN	Command FIFO underflow interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
6	CMD_FIFO_OVF_IEN	Command FIFO overflow interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
5	CMD_FIFO_THR_IEN	Command FIFO threshold interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
4	CMD_FIFO_EMPTY_IEN	Command FIFO empty interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
3	CMD_FIFO_FULL_IEN	Command FIFO full interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
2	SEQ_TIMEOUT_ERR_IEN	Sequencer timeout command error interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
1	SEQ_TIMEOUT_FINISHED_IEN	Sequencer timeout command finished. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
0	END_OF_SEQ_IEN	End of sequence interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Data FIFO Interrupt Enable Register

Address: 0x40080098, Reset: 0x00000000, Name: AFE_DATA_FIFO_IEN

Table 511. Bit Descriptions for AFE_DATA_FIFO_IEN

Bits	Bit Name	Description	Reset	Access
[31:5]	RESERVED	Reserved.	0x0000000	R
4	DATA_FIFO_UDF_IEN	Data FIFO underflow interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
3	DATA_FIFO_OVF_IEN	Data FIFO overflow interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
2	DATA_FIFO_THR_IEN	Data FIFO threshold interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
1	DATA_FIFO_EMPTY_IEN	Data FIFO empty interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
0	DATA_FIFO_FULL_IEN	Data FIFO full interrupt enable. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Analog Capture Interrupt Register

Address: 0x400800A0, Reset: 0x00000000, Name: AFE_ANALOG_CAPTURE_INT

The bits in this register are sticky when set. Each bit is cleared by writing a 1 to its location. Writing a 0 has no effect. If simultaneously the interrupt source is asserted and the core is attempting to clear a bit, the interrupt remains set.

A 1 means the interrupt source has been asserted since the last time the bit was cleared. A 0 means the interrupt source has not been asserted since the last time the bit was cleared.

Table 512. Bit Descriptions for AFE_ANALOG_CAPTURE_INT

Bits	Bit Name	Description	Reset	Access
[31:7]	RESERVED	Reserved.	0x0000000	R
6	ADC_DELTA_FAIL	ADC delta value check fail. When set, indicates that the difference between two consecutive ADC results was greater than the value specified by AFE_ ADCDELTA. If this bit is clear, it indicates that no difference between two consecutive ADC values greater than the limit has been detected since the last time this bit was cleared.	0x0	RW1C
5	ADC_MAX_FAIL	ADC maximum value check fail. When set, indicates that an ADC result was above the maximum value as specified by AFE_ADCMAX. If this bit is clear it indicates that no ADC value above the limit has been detected since the last time this bit was cleared.	0x0	RW1C
4	ADC_MIN_FAIL	ADC minimum value check fail. When set, indicates that an ADC result was below the minimum value as specified by AFE_ADCMIN. If this bit is clear it indicates that no ADC value below the limit has been detected since the last time this bit was cleared.	0x0	RW1C
3	TEMP_RESULT_READY	Temperature measurement result ready interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
2	SUPPLY_LPF_RESULT_READY	Supply rejection filter result ready interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
1	DFT_RESULT_READY	DFT result ready interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C

Bits	Bit Name	Description	Reset	Access
0	ADC_RESULT_READY	ADC result ready interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C

Analog Generation Interrupt Register

Address: 0x400800A4, Reset: 0x00000000, Name: AFE_ANALOG_GEN_INT

The bits in this register are sticky when set. Each bit is cleared by writing a 1 to its location. Writing a 0 has no effect. If simultaneously the interrupt source is asserted and the core is attempting to clear a bit, the interrupt remains set.

A 1 means the interrupt source has been asserted since the last time the bit was cleared. A 0 means the interrupt source has not been asserted since the last time the bit was cleared.

Table 513. Bit Descriptions for AFE_ANALOG_GEN_INT

Bits	Bit Name	Description	Reset	Access
[31:4]	RESERVED	Reserved.	0x00	R
3	CUSTOM_INT	General-purpose custom interrupt. Set manually through sequencer program. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
2	BREAK_SEQUENCE_ORG	Provides interrupt at fixed time intervals, to be used for sample sufficiency and drop detect steps. Set manually through sequencer programming. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
1	HARDWARE_SETUP_DONE	Indicates the MMR setup for the measurement step has finished, providing a hook for checking the MMRs. Set manually through sequencer programming. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
0	DELAY_COMMAND_DONE	Provides an early indication for the end of the test block. Set manually through sequencer programming. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C

Command FIFO Interrupt Register

Address: 0x400800A8, Reset: 0x00000010, Name: AFE_CMD_FIFO_INT

The bits in this register are sticky when set. Each bit is cleared by writing a 1 to its location. Writing a 0 has no effect. If simultaneously the interrupt source is asserted and the core is attempting to clear a bit, the interrupt remains set.

A 1 means the interrupt source has been asserted since the last time the bit was cleared. A 0 means the interrupt source has not been asserted since the last time the bit was cleared.

Table 514. Bit Descriptions for AFE_CMD_FIFO_INT

Bits	Bit Name	Description	Reset	Access
[31:8]	RESERVED	Reserved.	0x00	R
7	CMD_FIFO_UDF	Command FIFO underflow interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
6	CMD_FIFO_OVF	Command FIFO overflow interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
5	CMD_FIFO_THR	Command FIFO threshold interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
4	CMD_FIFO_EMPTY	Command FIFO empty interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x1	RW1C

Bits	Bit Name	Description	Reset	Access
3	CMD_FIFO_FULL	Command FIFO full interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
2	SEQ_TIMEOUT_ERR	Sequencer timeout command error. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
1	SEQ_TIMEOUT_FINISHED	Sequencer Timeout command finished. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
0	END_OF_SEQ	End of sequence interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C

Data FIFO Interrupt Register

Address: 0x400800AC, Reset: 0x00000002, Name: AFE_DATA_FIFO_INT

The bits in this register are sticky when set. Each bit is cleared by writing a 1 to its location. Writing a 0 has no effect. If simultaneously the interrupt source is asserted and the core is attempting to clear a bit, the interrupt remains set.

A 1 means the interrupt source has been asserted since the last time the bit was cleared. A 0 means the interrupt source has not been asserted since the last time the bit was cleared.

Table 515. Bit Descriptions for AFE_DATA_FIFO_INT

Bits	Bit Name	Description	Reset	Access
[31:5]	RESERVED	Reserved.	0x0000000	R
4	DATA_FIFO_UDF	Data FIFO underflow interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
3	DATA_FIFO_OVF	Data FIFO overflow interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
2	DATA_FIFO_THR	Data FIFO threshold interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C
1	DATA_FIFO_EMPTY	Data FIFO empty interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x1	RW1C
0	DATA_FIFO_FULL	Data FIFO full interrupt. 0: interrupt not asserted. 1: interrupt asserted.	0x0	RW1C

Switch Matrix Status (MSB) Register

Address: 0x400800B0, Reset: 0x00010000, Name: AFE_SW_STATUS_MSB

This register allows individual control of the AFE switches. The bit names are the same as the switch names in the AFE diagram.

Table 516. Bit Descriptions for AFE_SW_STATUS_MSB

Bits	Bit Name	Description	Reset	Access
[31:17]	RESERVED	Reserved.	0x0	R
16	PL_STATUS	PL switch control. 0: switch open. 1: switch closed.	0x1	R
15	P8_STATUS	P8 switch control. 0: switch open. 1: switch closed.	0x0	R
14	P7_STATUS	P7 switch control. 0: switch open. 1: switch closed.	0x0	R
13	P6_STATUS	P6 switch control. 0: switch open. 1: switch closed.	0x0	R
12	P5_STATUS	P5 switch control. 0: switch open. 1: switch closed.	0x0	R
11	P4_STATUS	P4 switch control. 0: switch open. 1: switch closed.	0x0	R
10	P3_STATUS	P3 switch control. 0: switch open. 1: switch closed.	0x0	R
9	P2_STATUS	P2 switch control. 0: switch open. 1: switch closed.	0x0	R
8	PR1_STATUS	PR1 switch control. 0: switch open. 1: switch closed.	0x0	R
7	D8_STATUS	D8 switch control. 0: switch open. 1: switch closed.	0x0	R
6	D7_STATUS	D7 switch control. 0: switch open. 1: switch closed.	0x0	R
5	D6_STATUS	D6 switch control. 0: switch open. 1: switch closed.	0x0	R
4	D5_STATUS	D5 switch control. 0: switch open. 1: switch closed.	0x0	R
3	D4_STATUS	D4 switch control. 0: switch open. 1: switch closed.	0x0	R
2	D3_STATUS	D3 switch control. 0: switch open. 1: switch closed.	0x0	R

Bits	Bit Name	Description	Reset	Access
1	D2_STATUS	D2 switch control. 0: switch open. 1: switch closed.	0x0	R
0	DR1_STATUS	DR1 switch control. 0: switch open. 1: switch closed.	0x0	R

Switch Matrix Status (LSB) Register

Address: 0x400800B4, Reset: 0x00010000, Name: AFE_SW_STATUS_LSB

This register allows individual control of the AFE switches. The bit names are the same as the switch names in the AFE diagram.

Table 517. Bit Descriptions for AFE_SW_STATUS_LSB

Bits	Bit Name	Description	Reset	Access
[31:18]	RESERVED	Reserved.	0x0	R
17	IVS_STATUS	IVS switch control. 0: switch open. 1: switch closed.	0x0	R
16	NL_STATUS	NL switch control. 0: switch open. 1: switch closed.	0x1	R
15	NR2_STATUS	NR2 switch control. 0: switch open. 1: switch closed.	0x0	R
14	N7_STATUS	N7 switch control. 0: switch open. 1: switch closed.	0x0	R
13	N6_STATUS	N6 switch control. 0: switch open. 1: switch closed.	0x0	R
12	N5_STATUS	N5 switch control. 0: switch open. 1: switch closed.	0x0	R
11	N4_STATUS	N4 switch control. 0: switch open. 1: switch closed.	0x0	R
10	N3_STATUS	N3 switch control. 0: switch open. 1: switch closed.	0x0	R
9	N2_STATUS	N2 switch control. 0: switch open. 1: switch closed.	0x0	R
8	N1_STATUS	N1 switch control. 0: switch open. 1: switch closed.	0x0	R
7	TR2_STATUS	TR2 switch control. 0: switch open. 1: switch closed.	0x0	R
6	T7_STATUS	T7 switch control. 0: switch open. 1: switch closed.	0x0	R

Bits	Bit Name	Description	Reset	Access
5	T6_STATUS	T6 switch control. 0: switch open. 1: switch closed.	0x0	R
4	T5_STATUS	T5 switch control. 0: switch open. 1: switch closed.	0x0	R
3	T4_STATUS	T4 switch control. 0: switch open. 1: switch closed.	0x0	R
2	T3_STATUS	T3 switch control. 0: switch open. 1: switch closed.	0x0	R
1	T2_STATUS	T2 switch control. 0: switch open. 1: switch closed.	0x0	R
0	T1_STATUS	T1 switch control. 0: switch open. 1: switch closed.	0x0	R

ADC Minimum Value Check Register

Address: 0x400800B8, Reset: 0x00000000, Name: AFE_ADCMIN

Table 518. Bit Descriptions for AFE_ADCMIN

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	ADC_MIN	ADC result minimum is configured by the user and is the lowest expected value to be measured by the ADC. If a value less than ADC_MIN is measured by the ADC, the ADC_MIN_FAIL bit is set.	0x0	RW

ADC Maximum Value Check Register

Address: 0x400800BC, Reset: 0x00000000, Name: AFE_ADCMAX

Table 519. Bit Descriptions for AFE_ADCMAX

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	ADC_MAX	ADC result maximum is configured by the user and is the highest expected value to be measured by the ADC. If a value greater than ADC_MAX is measured by the ADC the ADC_MAX_FAIL bit is set.	0x0	RW

ADC Delta Check Register

Address: 0x400800C0, Reset: 0x00000000, Name: AFE_ADCDELTA

Table 520. Bit Descriptions for AFE_ADCDELTA

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	Reserved.	0x0	R
[15:0]	ADC_DELTA	ADC result delta is configured by the user and is the greatest absolute difference two successive ADC measurements are expected to have. If a delta value greater than ADC_DELTA is measured by the ADC the ADC_DELTA_FAIL bit is set.	0x0	RW

Calibration Data Lock Register

Address: 0x40080100, Reset: 0x00000000, Name: AFE_CAL_DATA_LOCK

Table 521. Bit Descriptions for AFE_CAL_DATA_LOCK

Bits	Bit Name	Description	Reset	Access
[31:0]	CAL_DATA_PW	Password for calibration data registers, prevents overwriting of data after the calibration phase. 0xDE87A5AF: calibration data registers read/write. 0x00000000 to 0xDE87A5AE: calibration data registers read only. 0xDE87A5B0 to 0xFFFFFFFF: calibration data registers read only.	0x00000000	RW

ADC Gain (TIA Measurement) Register

Address: 0x40080104, Reset: 0x00004000, Name: AFE_ADC_GAIN_TIA

ADC gain value. An explanation of how this is used can be found in the ADC Channel section.

Table 522. Bit Descriptions for AFE_ADC_GAIN_TIA

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x0	R
[14:0]	ADC_GAIN_TIA	ADC gain correction for TIA measurement mode, stored as an unsigned number. Bit 14 represents the integer part and Bits[13:0] represent the fractional part. Allows a gain correction resolution of less than ±0.01%.	0x4000	RW

ADC Offset (TIA Measurement) Register

Address: 0x40080108, Reset: 0x00000000, Name: AFE_ADC_OFFSET_TIA

ADC TIA offset value. An explanation of how this is used can be found in the ADC Channel section.

Table 523. Bit Descriptions for AFE_ADC_OFFSET_TIA

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x00000	R
[14:0]	ADC_OFFSET_TIA	ADC offset correction in TIA measurement mode, represented as a twos complement number. It allows a ±5% correction range with a resolution of 0.25 LSBs at 16-bit level. 0x3FFF: +4095.75 0x0001: +0.25 0x0000: 0 0x7FFF: −0.25 0x4000: −4096.0	0x000	RW

ADC Gain (Temperature Sensor Measurement) Register

Address: 0x4008010C, Reset: 0x00004000, Name: AFE_ADC_GAIN_TEMP_SENS

ADC gain value, used when measuring the temperature when using the internal temperature sensor. An explanation of how this is used can be found in the ADC Channel section.

Table 524. Bit Descriptions for AFE_ADC_GAIN_TEMP_SENS

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x0	R
[14:0]	ADC_GAIN_TEMP_SENS	ADC gain correction for temperature measurement mode, stored as an unsigned number. Bit 14 represents the integer part and Bits[13:0] represent the fractional part. Allows a gain correction resolution of less than ±0.01%.	0x4000	RW

ADC Offset (Temperature Sensor Measurement) Register

Address: 0x40080110, Reset: 0x00000000, Name: AFE_ADC_OFFSET_TEMP_SENS

ADC offset value, used when measuring the temperature when using the internal temperature sensor. An explanation of how this is used can be found in the ADC Channel section.

Table 525. Bit Descriptions for AFE_ADC_OFFSET_TEMP_SENS

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x00000	R
[14:0]	ADC_OFFSET_TEMP_SENS	ADC offset correction in temperature measurement mode, represented as a twos complement number. It allows a ±5% correction range with a resolution of 0.25 LSBs at 16-bit level. 0x3FFF: +4095.75. 0x0001: +0.25. 0x0000: 0. 0x7FFF: −0.25. 0x4000: −4096.	0x000	RW

ADC Gain (Auxiliary Channel Measurement) Register

Address: 0x40080118, Reset: 0x00004000, Name: AFE_ADC_GAIN_AUX

Auxiliary channel gain value. An explanation of how this is used can be found in the ADC Channel section.

Table 526. Bit Descriptions for AFE_ADC_GAIN_AUX

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x0	R
[14:0]	ADC_GAIN_AUX	ADC gain correction for auxiliary channel measurement mode, stored as an unsigned number. Bit 14 represents the integer part and Bits[13:0] represent the fractional part. Allows a gain correction resolution of less than ±0.01%.	0x4000	RW

ADC Offset (Auxiliary Channel Measurement) Register

Address: 0x4008011C, Reset: 0x00000000, Name: AFE_ADC_OFFSET_AUX

ADC auxiliary channel offset value. An explanation of how this is used can be found in the ADC Channel section.

Table 527. Bit Descriptions for AFE_ADC_OFFSET_AUX

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x0	R
[14:0]	ADC_OFFSET_AUX	ADC offset correction in auxiliary channel measurement mode, represented as a twos complement number. It allows a ±5% correction range with a resolution of 0.25 LSBs at 16-bit level. 0x3FFF: +4095.75. 0x0001: +0.25. 0x0000: 0. 0x7FFF: −0.25. 0x4000: −4096.	0x0	RW

DAC Offset with Attenuator Disabled Register

Address: 0x40080120, Reset: 0x00000000, Name: AFE_DAC_OFFSET_UNITY

Table 528. Bit Descriptions for AFE_DAC_OFFSET_UNITY

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	R
[11:0]	DAC_OFFSET_UNITY	DAC offset correction factor, signed number represented in twos complement format with a 0.5 LSB precision. Used when attenuator is disabled. 0x7FF: $2^{10} - 0.5$. 0x001: +0.5. 0x000: 0. 0xFFF: −0.5. 0x800: -2^{10}.	0x000	RW

DAC Offset with Attenuator Enabled Register

Address: 0x40080124, Reset: 0x00000000, Name: AFE_DAC_OFFSET_ATTEN

Table 529. Bit Descriptions for AFE_DAC_OFFSET_ATTEN

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	R
[11:0]	DAC_OFFSET_ATTEN	DAC offset correction factor, signed number represented in twos complement format with a 0.5 LSB precision. Used when attenuator is enabled. 0x7FF: $2^{10} - 0.5$. 0x001: +0.5. 0x000: 0. 0xFFF: −0.5. 0x800: -2^{10}.	0x000	RW

DAC Gain Register

Address: 0x40080128, Reset: 0x00000800, Name: AFE_DAC_GAIN

Table 530. Bit Descriptions for AFE_DAC_GAIN

Bits	Bit Name	Description	Reset	Access
[31:12]	RESERVED	Reserved.	0x00000	R
[11:0]	DAC_GAIN	DAC gain correction factor, unsigned number. 0x000: 0. 0x800: 1. 0xFFF: $2 - 1/(2^{11})$.	0x800	RW

Precision Reference Trim 0 Register

Address: 0x4008012C, Reset: 0x00000000, Name: AFE_REF_TRIM0

Analog Devices factory trim register for precision reference.

Table 531. Bit Descriptions for AFE_REF_TRIM0

Bits	Bit Name	Description	Reset	Access
[31:13]	RESERVED	Reserved.	0x0	R
[12:6]	VBIASABSTRIM	Analog Devices factory trim of absolute value of VBIAS.	0x0	RW
[5:0]	ITATTRIM	Analog Devices factory trim of absolute value of master bias current for AFE blocks.	0x0	RW

Precision Reference Trim 1 Register

Address: 0x40080130, Reset: 0x00000000, Name: AFE_REF_TRIM1

Analog Devices factory trim register for precision reference.

Table 532. Bit Descriptions for AFE_REF_TRIM1

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
[19:12]	REFTC	Analog Devices factory trim of temperature coefficient of VREF.	0x0	RW
[11:8]	REFCP	Analog Devices factory trim of crossing point of VREF.	0x0	RW
[7:0]	VREFABSTRIM	Analog Devices factory trim of absolute value of VREFADC.	0x0	RW

Analog LDO Trim Register

Address: 0x40080134, Reset: 0x00000000, Name: AFE_ALDO_TRIM

Analog Devices factory calibration register for analog LDO.

Table 533. Bit Descriptions for AFE_ALDO_TRIM

Bits	Bit Name	Description	Reset	Access
[31:4]	RESERVED	Reserved.	0x0	R
[3:0]	AVDDADJ	Analog Devices factory trim of AVDD voltage.	0x0	RW

DAC Trim Register

Address: 0x40080138, Reset: 0x00000000, Name: AFE_DAC_TRIM

Analog Devices factory calibration register for the DAC reconstruction filter.

Bypass of the RCF is achieved by putting DACRCFOFFSETEN to 0 and DACRCFCAL[5:0] to 0x00.

The RCF is implemented with a Series R and a bank of parallel caps, each cap with a series switch to ground. Bypass mode is done by opening all switches by putting DACRCFOFFSETEN to 0 and DACRCFCAL[5:0] to 0x00. In normal operation (RCF not bypassed), DACRCFOFFSETEN is 1 to enable a large offset cap. The required amount of remaining smaller caps in the bank are enabled to get the correct corner frequency.

Table 534. Bit Descriptions for AFE_DAC_TRIM

Bits	Bit Name	Description	Reset	Access
[31:9]	RESERVED	Reserved.	0x000000	R
8	DACRCFOFFSETEN	Used only when the RCF needs to be bypassed. In normal operation this is set to 1. 0: RCF disabled and bypassed. 1: RCF enabled, normal operation.	0x0	RW
[7:6]	RESERVED	Reserved.	0x0	R
[5:0]	DACRCFCAL	Analog Devices factory trim of RCF corner frequency.	0x00	RW

INAMP Trim Register

Address: 0x4008013C, Reset: 0x00000000, Name: AFE_INAMP_TRIM

ADI factory calibration register for INAMP.

Table 535. Bit Descriptions for AFE_INAMP_TRIM

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x00000	R
[14:6]	EXCITELOOPCONFIG	Analog Devices factory configuration of excitation loop.	0x000	RW
[5:0]	INAMPOFFSETADJ	Analog Devices factory trim of excitation loop offset.	0x00	RW

Excitation Buffer Trim Register

Address: 0x40080140, Reset: 0x00000000, Name: AFE_EXBUF_TRIM

Analog Devices factory calibration register for excitation buffer.

Table 536. Bit Descriptions for AFE_EXBUF_TRIM

Bits	Bit Name	Description	Reset	Access
[31:17]	RESERVED	Reserved.	0x0	R
[16:11]	EXBUFZEROCONTROL	Analog Devices factory trim of excite buffer open-loop zero.	0x0	RW
[10:5]	EXBUFGAINCONTROL	Analog Devices factory trim of excitation loop gain bandwidth.	0x0	RW
[4:0]	EXBUFOFFSETADJ	Unused.	0x0	RW

Temperature Sensor Trim Register

Address: 0x40080144, Reset: 0x00004000, Name: AFE_TEMP_SENS_TRIM

Table 537. Bit Descriptions for AFE_TEMP_SENS_TRIM

Bits	Bit Name	Description	Reset	Access
[31:15]	RESERVED	Reserved.	0x00000	R
[14:0]	TEMP_SENS_GAIN	Gain correction factor for the temperature sensor, calculated during factory calibration.	0x4000	RW

NO FACTORY CALIBRATION

INTRODUCTION

The No Factory Calibration section, along with the subsequent AFE Example Use Cases section, documents how to calibrate and make measurements using the analog front end. The routines clarify the sequence of functions, the required register names, and the bit fields that need to be enabled/disabled to execute the example.

PROPOSED CALIBRATION ROUTINE FOR AUXILIARY CHANNEL

For accurate measurement of the auxiliary channels, including external thermistor and internal supply voltages, a calibration routine must be performed each time the AFE is powered up.

The calibration routine for the AFE is as described in Table 538 to Table 540 (sequencer control is omitted, for example, DMA setup, start, and end of sequences). Wait times indicated are for illustration purposes only.

Table 538. Calibration Routine for AFE

Action	Register	Bit	Value	Comments
Enable Analog LDO and Analog LDO Current Limit	AFE_CFG	ALDO_EN	1	
		ALDOILIMIT_EN	1	
Wait 2 ms	Not applicable	Not applicable	0x00007D00	Time to charge 0.47 µF capacitor to 1.8 V when current limited.
Disable Analog LDO Current Limit, Enable VREF and VREF Current Limit	AFE_CFG	ALDOILIMIT_EN	0	
		REF_EN	1	
		VREFBUFILIMIT_EN	1	
Wait 20 ms	Not applicable	Not applicable	0x0004E200	Time to charge 4.7 µF capacitor to 1.8 V when current limited.
Disable VREF Current Limit and Enable VBIAS	AFE_CFG	VREFBUFILIMIT_EN	0	
		VBIASBUF_EN	1	
Wait 5 ms	Not applicable	Not applicable	0x00013880	Time to charge 0.47 µF capacitor to 1.1 V.
Enable ADC	AFE_CFG	ADC_EN	1	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow power up of ADC circuitry.
Set ADC Mux to Measure Auxiliary Channel Offset (ADC_OFFSET_AUX)	AFE_ADC_CFG	MUX_SEL	11100	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN	1	
		SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YA, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
Set ADC Mux to Measure a Reference Voltage (ADC_GAIN_AUX)	AFE_ADC_CFG	MUX_SEL	11101	

Action	Register	Bit	Value	Comments
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YB, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Offset Correction for Auxiliary Channel Measurement Mode (Executed by Cortex-M3)	AFE_ADC_OFFSET_AUX	ADC_OFFSET_AUX	Write 4× OFFSET	Offset correction factor: OFFSET = −20,025/GAIN − Ya
Set ADC Gain Correction for Auxiliary Channel Measurement Mode (Executed by Cortex-M3)	AFE_ADC_GAIN_AUX	ADC_GAIN_AUX	16,384× GAIN	Gain correction factor: GAIN = $(2^{15})/(Yb - Ya)$

Because the auxiliary channel is now calibrated, the following can be performed accurately (the user has the option to do either step first).

Table 539. Example Temperature Measurement Using the External Thermistor

Action	Register	Bit	Value	Comments
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
Set ADC Mux to Measure an AN_A (Assumes Auxiliary Channel with Thermistor in on AN_A)	AFE_ADC_CFG	MUX_SEL	01000	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YAN_A, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Calculate Voltage on AN_A in Volts, VAN_A (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	VANA_A = $3.6(YAN\_A/(2^{16})) + 1.1$
Use VANA_A to Derive the Temperature of the Thermistor (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Decide if Measurement Is to Proceed	Not applicable	Not applicable	Not applicable	

Table 540. Example Battery Measurement (VCCM_ANA)

Action	Register	Bit	Value	Comments
Set ADC Mux to Measure VCCM_ANA	AFE_ADC_CFG	MUX_SEL	10000	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YVCCM_ANA, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Calculate the VCCM_ANA Voltage in Volts, VVCCM_ANA(Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	VVCCM_ANA = 2(3.6(YVCCM_ANA/(2^{16})) + 1.1) (This equation applies to measurement of any of the supply voltage channels.)
Decide if Measurement Is to Proceed	Not applicable	Not applicable	Not applicable	

At this stage, it has been decided whether to proceed with the sensor measurement. If a sensor measurement is to occur, the ADC TIA channel must be calibrated. An error analysis of this calibration and the routine to perform the calibration is summarized in the Proposed Calibration Routine for the Current Measurement Through TIA section.

Note that the offset and gain registers for the ADC auxiliary channel measurements are not retained when the device is in hibernate mode; these registers must be set as 1 (AFE_ADC_GAIN_AUX) and 0 (AFE_ADC_OFFSET_AUX) in the automatic calibration routine.

PROPOSED CALIBRATION ROUTINE FOR THE CURRENT MEASUREMENT THROUGH TIA

In this section, we assume a ratiometric configuration measuring the current through an RCAL and subsequently a sensor.

Set up an accurate current source using a recommended external 0.1% tolerance 25 ppm/°C RCAL resistor and the accurate on-chip 25 ppm/°C reference voltage. This accurate current source can then be used to achieve the goal of removing gain and offset errors in the transimpedance amplifier and the ADC lineup. After these errors are removed, the dominating system errors come from the internal reference tolerance and drift. The values in Table 541 are added using an rms method due to the random and uncorrelated nature.

Table 541. System Measurement Error After Automatic Calibration Routine

	Measurement Error	
Specification	**mV**	**% Error**
Maximum VREF Tolerance	±3 mV in 1.8 V	±0.166%
VREF Temperature Variation (±48 ppm/°C × ±25°C)	±2.16 mV	±0.12%
Calibration Resistor Tolerance		±0.1%
Calibration Resistor Temperature Variation (±25 ppm/°C × ±25°C)		±0.0625%
Total System Error from 0°C to 50°C (Add in RMS Noise Sense)		±0.236%

NO FACTORY CALIBRATION SEQUENCE

For this routine, VBIAS = 1.1 V and VREF = 1.8 V.

Step 1: AFE Power-Up Sequence

Turn on the DAC, excitation buffer, in amp, and TIA, and set the switch matrix to excite the calibration resistor. This assumes that the analog LDO, precision reference, and ADC have already been enabled. See the Proposed Calibration Routine for Auxiliary Channel section describing the calibration of the auxiliary channel and the temperature measurement using the auxiliary channel.

Table 542. No Factory Calibration Sequence Step 1

Action	Register	Bit	Value	Comments
Disabled DAC Attenuator	AFE_DAC_CFG	DAC_ATTEN_EN	0	
Set Switch Matrix to All Open State	AFE_SW_CFG	DMUX_STATE PMUX_STATE NMUX_STATE TMUX_STATE	0000 0000 0000 0000	
Enable DAC/RCF/ATTEN, TIA, INAMP, and Excitation Buffer	AFE_CFG	DAC_EN TIA_EN INAMP_EN BUF_EN	1 1 1 1	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow power-up of DAC, TIA, INAMP, and EXBUF circuitry.
Set Switch Matrix to AC RCAL State	AFE_SW_CFG	AFE_SW_CFG[17:0]	0x08811	SW_SOURCE_SEL = 0
Program DAC to apply ~90 mV DC Across RCAL	AFE_WG_DAC_CODE	DAC_CODE	0x8E7	

Step 2: Measure ADC Offset Error

Measure the ADC offset error. Apply known 0 mV voltage to ADC channel input. This is achieved by applying (VBIAS, VBIAS) and taking the ADC result.

Table 543. No Factory Calibration Sequence Step 2

Action	Register	Bit	Value	Comments
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
Set ADC Mux to Measure the ADC Channel Offset in Sensor Measurement Mode (AFE_ADC_OFFSET_TIA)	AFE_ADC_CFG	MUX_SEL	11000	Set both differential inputs of the ADC channel to VBIAS to measure offset.
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YA, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Offset Correction for TIA Measurement Mode (Executed by Cortex-M3)	AFE_ADC_OFFSET_TIA	ADC_OFFSET_TIA	4× OFFSET1	Offset correction factor: OFFSET1 = –YA.

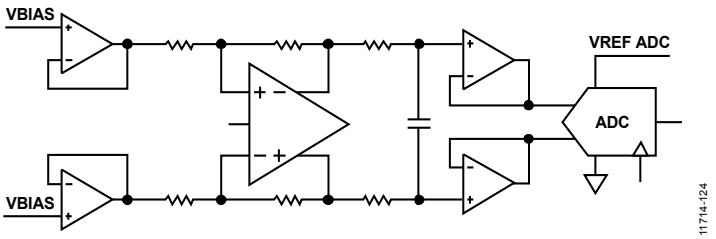


Figure 125. Measure ADC Offset, Simplified Diagram

Step 3: Measure ADC Gain Error

Measure ADC gain error. Apply known 0.7 V to ADC channel input. This is achieved by applying (VREF, VBIAS) and taking the ADC result.

Table 544. No Factory Calibration Sequence Step 3

Action	Register	Bit	Value	Comments
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
Set ADC Mux to Measure the ADC Channel Gain in Sensor Measurement Mode (AFE_ADC_GAIN_TIA)	AFE_ADC_CFG	MUX_SEL	11001	ADC mux to (VREF, VBIAS) = 0.7 V
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YB, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Gain Correction for TIA Measurement Mode (Executed by Cortex-M3)	AFE_ADC_GAIN_TIA	ADC_GAIN_TIA	16,384 × GAIN1	Gain correction factor: GAIN1 = (0.7/3.6)(2^{16})/YB

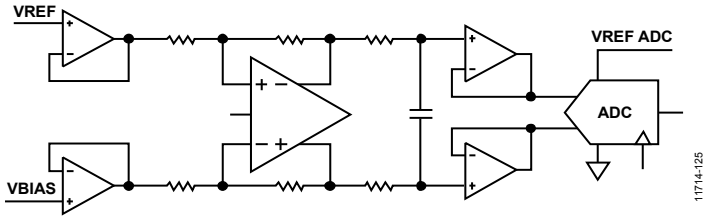


Figure 126. Measure ADC Gain Error, Simplified Diagram

Step 4: Determine I_{CAL}

The following assumes that RCAL = 1 kΩ. Set the DAC to put ~90 mV across the calibration resistor. This sets up a reference current of approximately 90 µA (90% full scale) that is used for calibration. Set the ADC mux to select P and N as the ADC inputs. Accurately measure the voltage that has been applied across the calibration resistor using the ADC.

Table 545. No Factory Calibration Sequence Step 4

Action	Register	Bit	Value	Comments
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
ADC Mux to (P, N); Set ADC Mux to Accurately Measure the Voltage Across the Calibration Resistor (P/N Automatic Calibration)	AFE_ADC_CFG	MUX_SEL	11110	

Action	Register	Bit	Value	Comments
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YRES, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	

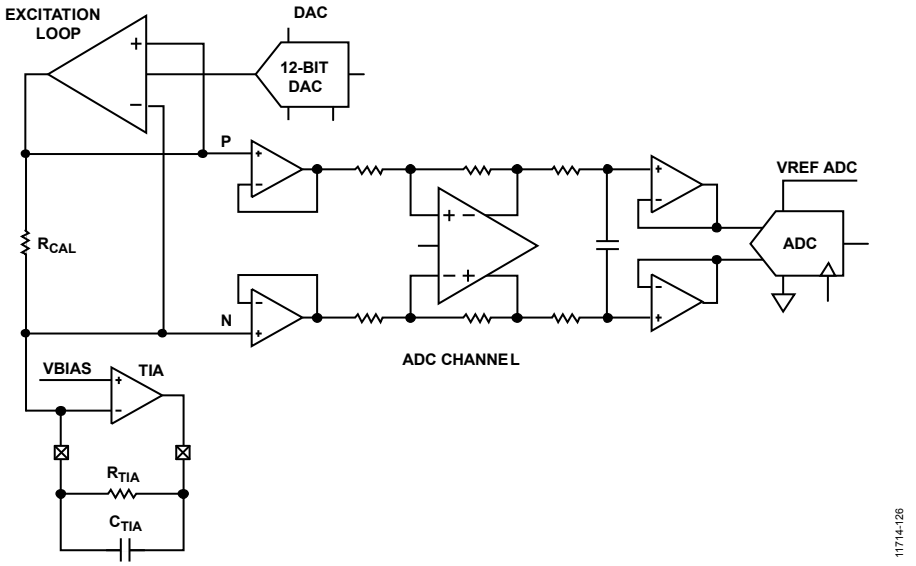


Figure 127. Determine I_{CAL}

Step 5: Measure TIA Offset

Program the switch matrix to dc calibration open state. Expected input current to TIA is 0 A.

Table 546. No Factory Calibration Sequence Step 5

Action	Register	Bit	Value	Comments
Switch Matrix to Switch Cal Open State	AFE_SW_CFG	AFE_SW_CFG[17:0]	0x03000	
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
ADC Mux to TIA	AFE_ADC_CFG	MUX_SEL	00010	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YP, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	

Action	Register	Bit	Value	Comments
Set ADC Offset Correction for TIA Measurement Mode (Executed by Cortex-M3)	AFE_ADC_OFFSET_TIA	ADC_OFFSET_TIA	4× OFFSET2	Offset correction factor: OFFSET2 = OFFSET1 − Yp/GAIN1

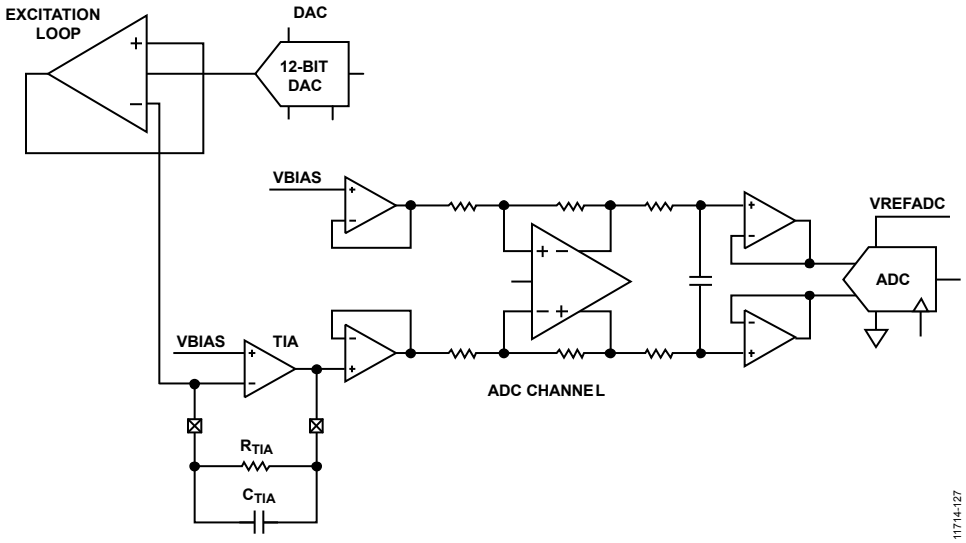


Figure 128. Measure TIA Offset, Simplified Diagram

Step 6: Calibrate R_{GAIN}

Switch the ADC mux to the current measurement channel, and make a current measurement through the ADC.

Table 547. No Factory Calibration Sequence Step 6

Action	Register	Bit	Value	Comments
Switch Matrix to AC RCAL State	AFE_SW_CFG	AFE_SW_CFG[17:0]	0x08811	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YQ, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Gain Correction for TIA Measurement Mode (Executed by Cortex-M3)	AFE_ADC_GAIN_TIA	ADC_GAIN_TIA	16,384 × GAIN2	Gain correction factor: GAIN2 = −GAIN1 × (7.5YRES)/YQ

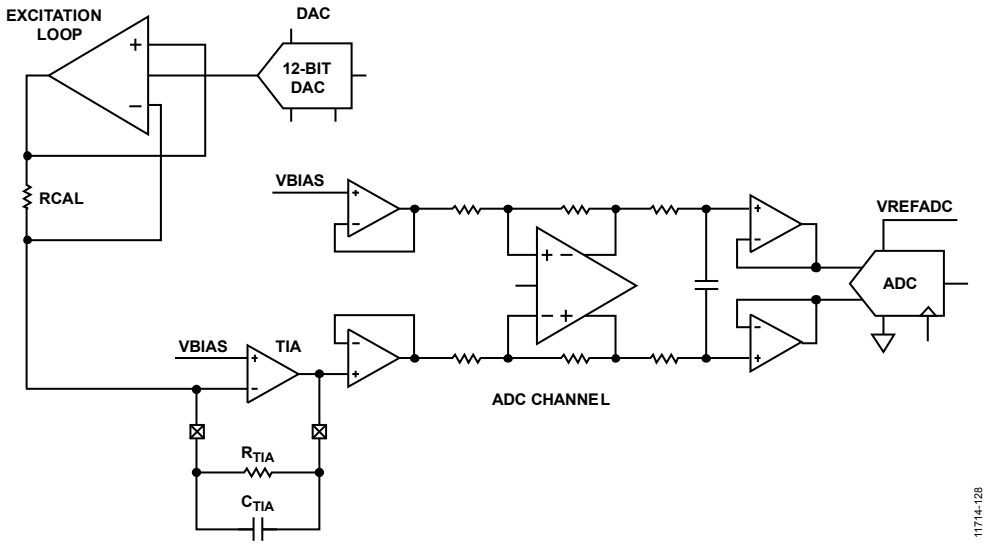


Figure 129. Calibrate R_{GAIN}, Simplified Diagram

Note that the offset and gain registers for the ADC TIA (sensor) channel measurements are not retained in hibernate mode and must be set as 1 (AFE_ADC_GAIN_TIA) and 0 (AFE_ADC_OFFSET_TIA) in the autocalibration routine.

PROPOSED CALIBRATION ROUTINE FOR INTERNAL TEMPERATURE SENSOR ON AFE POWER-UP

For accurate measurement of the temperature using the internal temperature sensor, a calibration routine must be performed when the AFE is powered up.

The calibration routine for the AFE is described in Table 548 and Table 549 (sequencer control is omitted, for example, DMA setup, start, and end of sequences). This routine assumes that the analog LDO, VREF, and VBIAS have already been enabled as previously described.

Table 548. Calibration Routine for AFE

Action	Register	Bit	Value	Comments
Enable ADC	AFE_CFG	ADC_EN	1	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow power-up of ADC circuitry.
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
Set ADC Mux to Measure Internal Temperature Sensor Channel Offset (ADC_OFFSET_TEMP_SENS)	AFE_ADC_CFG	MUX_SEL	11010	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN	1	
		SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YO, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Mux to Measure a Gain Calibration Point A	AFE_ADC_CFG	MUX_SEL	11000	

Action	Register	Bit	Value	Comments
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	Sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YA, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Mux to Measure a Gain Calibration Point B	AFE_ADC_CFG	MUX_SEL	11001	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Convert Filtered ADC Result to a Signed Number, YB, by Subtracting 32,768 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set ADC Offset Correction in Temperature Measurement Mode[1] (Executed by Cortex-M3)	AFE_ADC_OFFSET_TEMP_SENS	ADC_OFFSET_TEMP_SENS	4× offset	Offset correction factor: offset = −YO
Set ADC Gain Correction for Temperature Measurement Mode[1] (Executed by Cortex-M3)	AFE_ADC_GAIN_TEMP_SENS	ADC_GAIN_TEMP_SENS	16,384× gain	Gain correction factor: gain = 19,115/(YB − YA)

[1] The gain and offset results determined in this step are the same as the first two steps of the TIA channel calibration.

Because the internal temperature sensor channel is now calibrated, the sequence described in Table 549 can be done accurately.

Table 549. Temperature Measurement Using the Internal Temperature Sensor

Action	Register	Bit	Value	Comments
Enable Temperature Sensor	AFE_CFG	TEMP_SENSOR_EN	1	
Break-Before-Make in ADC Mux	AFE_ADC_CFG	MUX_SEL	00000	
Set ADC Mux to Measure an Internal Temperature Sensor	AFE_ADC_CFG	MUX_SEL	00011	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable Temperature Reading, ADC Conversion and Filters.	AFE_CFG	TEMP_CONV_EN, ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Read Temperature Sensor Result (Executed by Cortex-M3)	AFE_TEMP_SENSOR_RESULT	Temperature	Not Applicable	
Calculate the Temperature in °C, TDEGC (Executed by Cortex-M3)	Not Applicable	Not Applicable	Not Applicable	TDEGC = (AFE_TEMP_SENSOR_RESULT – 3,414)/12.5

Note that the offset and gain registers for the ADC temperature channel measurements are not retained when the device is in hibernate mode; these registers must be set as 1 (AFE_ADC_GAIN_TEMP_SENS) and 0 (AFE_ADC_OFFSET_TEMP_SENS) in the automatic calibration routine.

PROPOSED CALIBRATION ROUTINE FOR EXCITATION CHANNEL (DAC AND EXCITATION LOOP) ON THE AFE POWER-UP

For minimum dc offset on the sensor excitation voltage, it is recommended to perform an offset calibration routine before sensor excitation as described in this section. It is recommended to perform this calibration routine separately for both the attenuator mode and the nonattenuator mode.

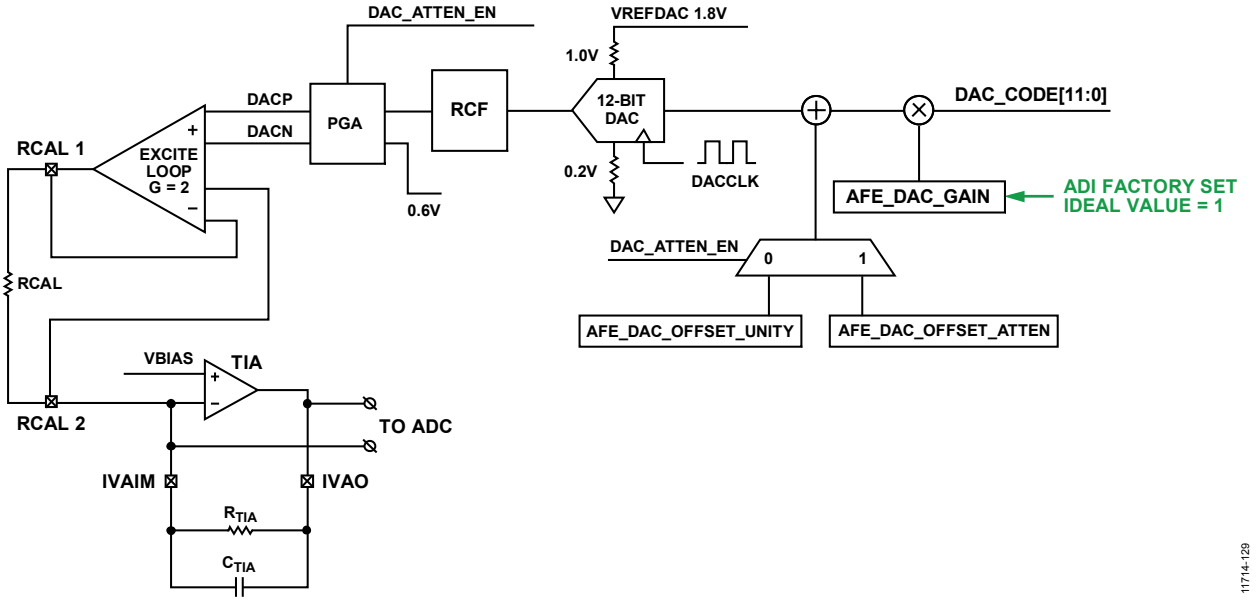


Figure 130. Excitation Architecture Showing Offset Correction Registers

Offset Calibration for Attenuator Mode

This calibration routine assumes that LDO, VREF, and VBIAS are the references; that ADC, DAC, INAMP, EXBUF, and TIA have already been enabled; and that the ADC TIA channel has already been calibrated. The Value 1 and Value 2 referred to in Table 550 are chosen to straddle 0 V dc relative to the current to voltage (I-V) input while being sufficiently small enough to prevent saturation of the I/V amp when it is applied across the RCAL calibration resistor.

Table 550. Offset Calibration Routine for Attenuation Mode—DC Offset Calibration

Action	Register	Bit	Value	Comments
Enable DAC Attenuation	AFE_DAC_CFG	DAC_ATTEN_EN	1	
Switch Matrix to AC RCAL State	AFE_SW_CFG	AFE_SW_CFG[17:0]	0x08811	
Set DAC to Midscale	AFE_WG_DAC_CODE	DAC_CODE	0x800	
Set Offset Correction to Value 1	AFE_DAC_OFFSET_ATTEN	DAC_OFFSET_ATTEN	0xA00	
ADC Mux to TIA	AFE_ADC_CFG	MUX_SEL	00010	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Store Result as Result 1 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set Offset Correction to Value 2	AFE_DAC_OFFSET_ATTEN	DAC_OFFSET_ATTEN	0x600	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Store Result as Result 2 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set Offset Correction (Executed by Cortex-M3)	AFE_DAC_OFFSET_ATTEN	DAC_OFFSET_ATTEN	Value depends on bit settings of the register	Using Result 1, Result 2, Value 1, and Value 2, interpolate to get offset correction.

Offset Calibration for Nonattenuator Mode

Table 551. Offset Calibration Routine for Nonattenuator Mode—DC Offset Calibration

Action	Register	Bit	Value	Comments
Disable DAC Attenuation	AFE_DAC_CFG	DAC_ATTEN_EN	0	
Switch Matrix to AC RCAL State	AFE_SW_CFG	AFE_SW_CFG[17:0]	0x08811	
Set DAC to Midscale	AFE_WG_DAC_CODE	DAC_CODE	0x800	
Set Offset Correction to Value 1	AFE_DAC_OFFSET_UNITY	DAC_OFFSET_UNITY	0xFB2	
ADC Mux to TIA	AFE_ADC_CFG	MUX_SEL	00010	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Store Result as Result 1 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set Offset Correction to Value 2	AFE_DAC_OFFSET_UNITY	DAC_OFFSET_UNITY	0x04E	
Wait 100 µs	Not applicable	Not applicable	0x00000640	Time to allow propagation and settling of newly selected voltage through antialias filter.
Enable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	1	
Wait 37 ms	Not applicable	Not applicable	0x00090880	sinc2hf and sinc2lf latency.
Disable ADC Conversion and Filters	AFE_CFG	ADC_CONV_EN, SUPPLY_LPF_EN	0	
Store Result as Result 2 (Executed by Cortex-M3)	Not applicable	Not applicable	Not applicable	
Set DAC Offset Correction Factor (Executed by Cortex-M3)	AFE_DAC_OFFSET_UNITY	DAC_OFFSET_UNITY	Value depends on bit settings of the register	Using Result 1, Result 2, Value 1, and Value 2, interpolate to get offset correction.

AFE EXAMPLE USE CASES

INTRODUCTION

This section documents suggested timing diagram examples to optimally use the analog front end. The examples clarify the sequence of functions, required register names, and bit fields that must be enabled/disabled to execute the example. The timing between particular register writes is also identified. Most of these steps are also covered in the No Factory Calibration section, which documents the actual register writes in more detail.

EXAMPLE USE CASE FLOW DIAGRAM

The flow diagram in Figure 131 highlights the sequence of the example use cases. The sequence is as follows:

1. AFE power-up
2. Auxiliary channel calibration
3. Auxiliary channel measurement
4. Internal temperature sensor channel calibration
5. Internal temperature sensor measurement
6. Excitation channel power-up
7. TIA channel calibration
8. Excitation channel calibration (attenuation enabled)
9. Excitation channel calibration (no attenuation)
10. AC measurement example
11. Attenuation change
12. DC measurement example
13. AFE power-down

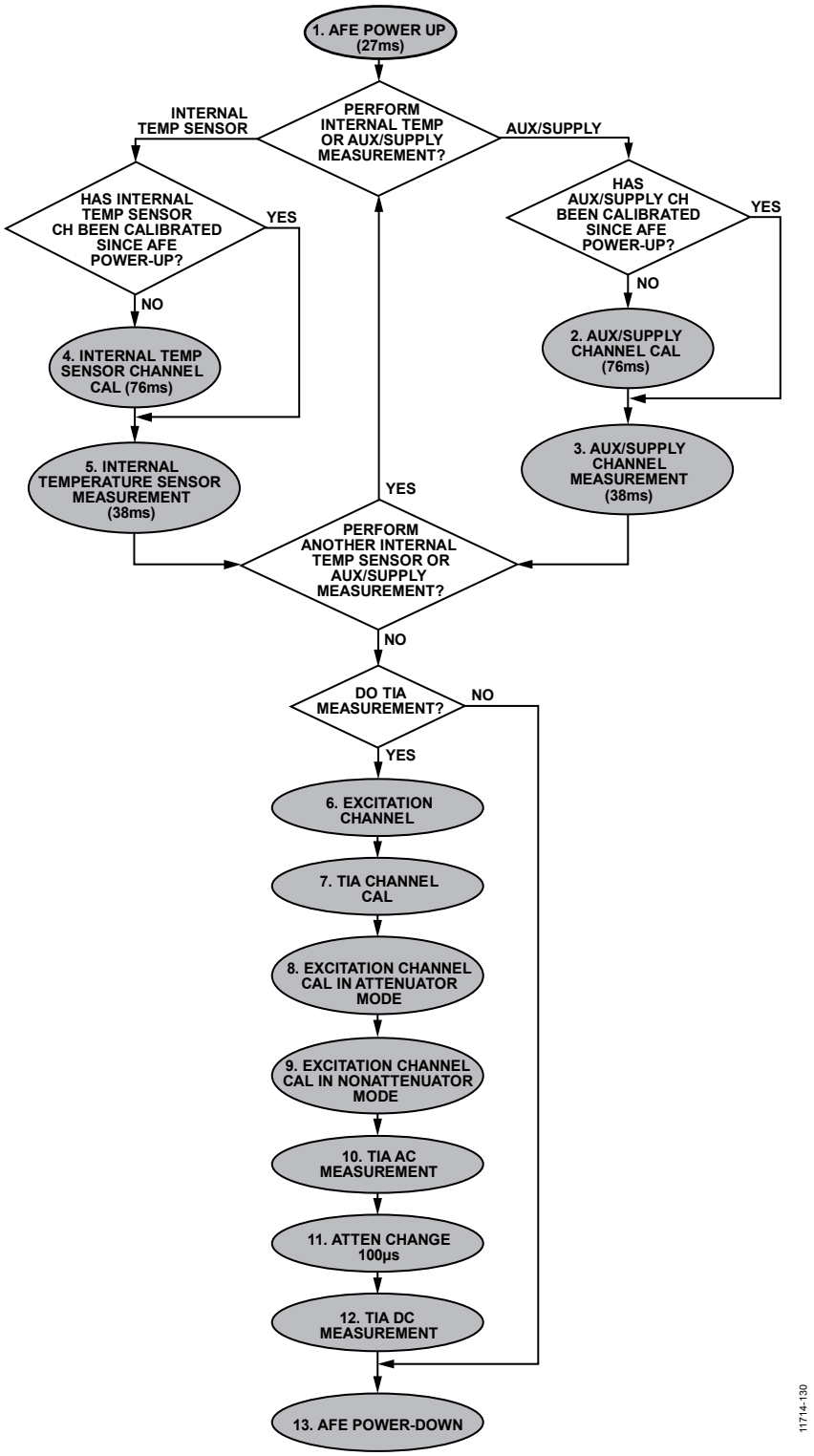
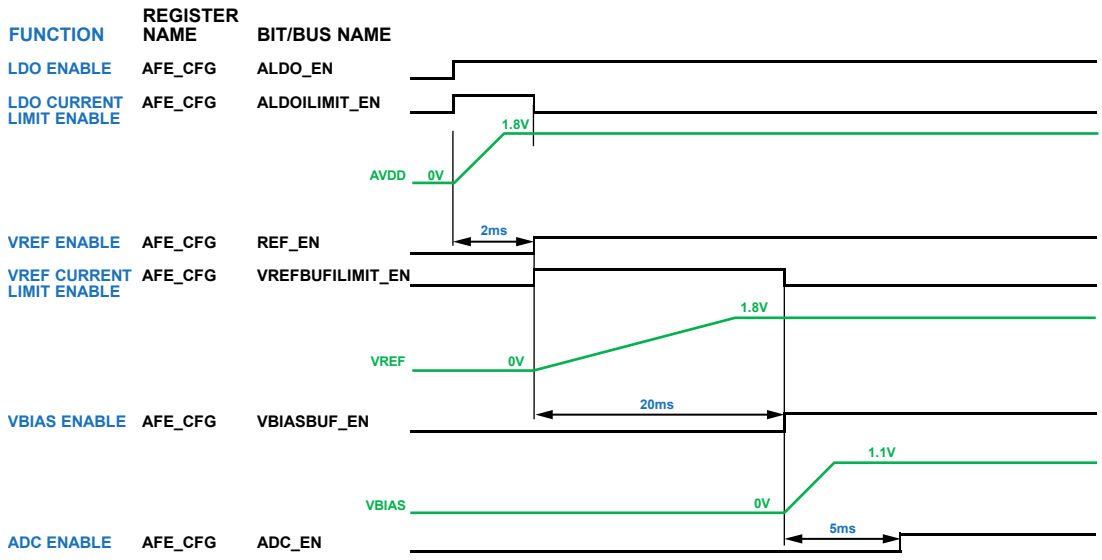


Figure 131. AFE Example Case Flow Diagram

Example Use Case Timing Diagrams

Step 1. AFE Power-Up



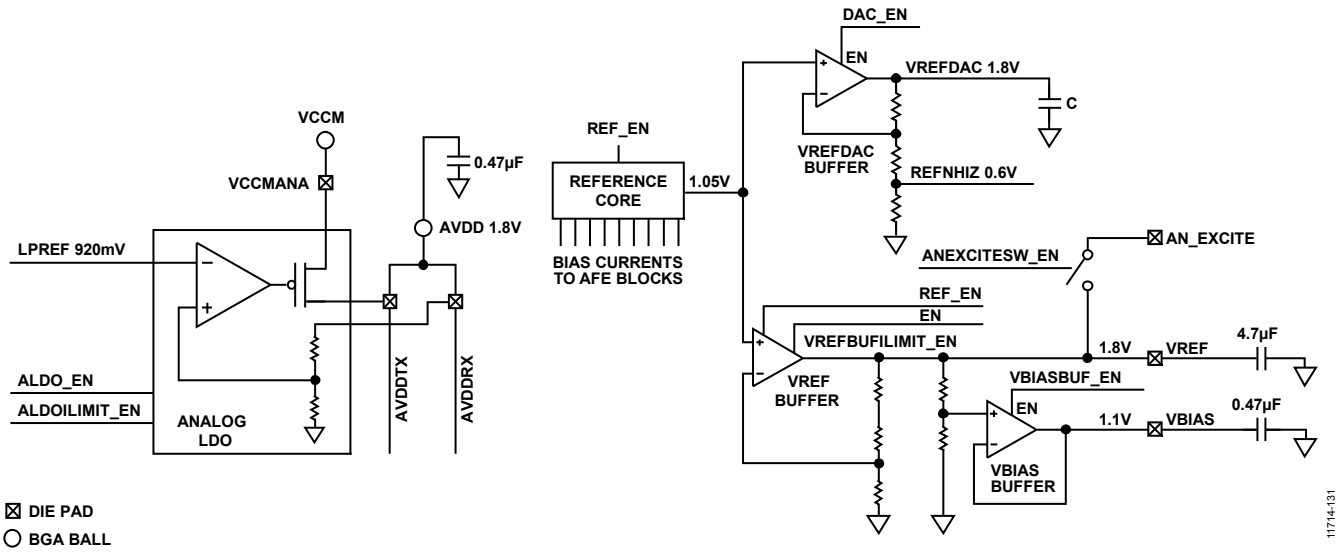
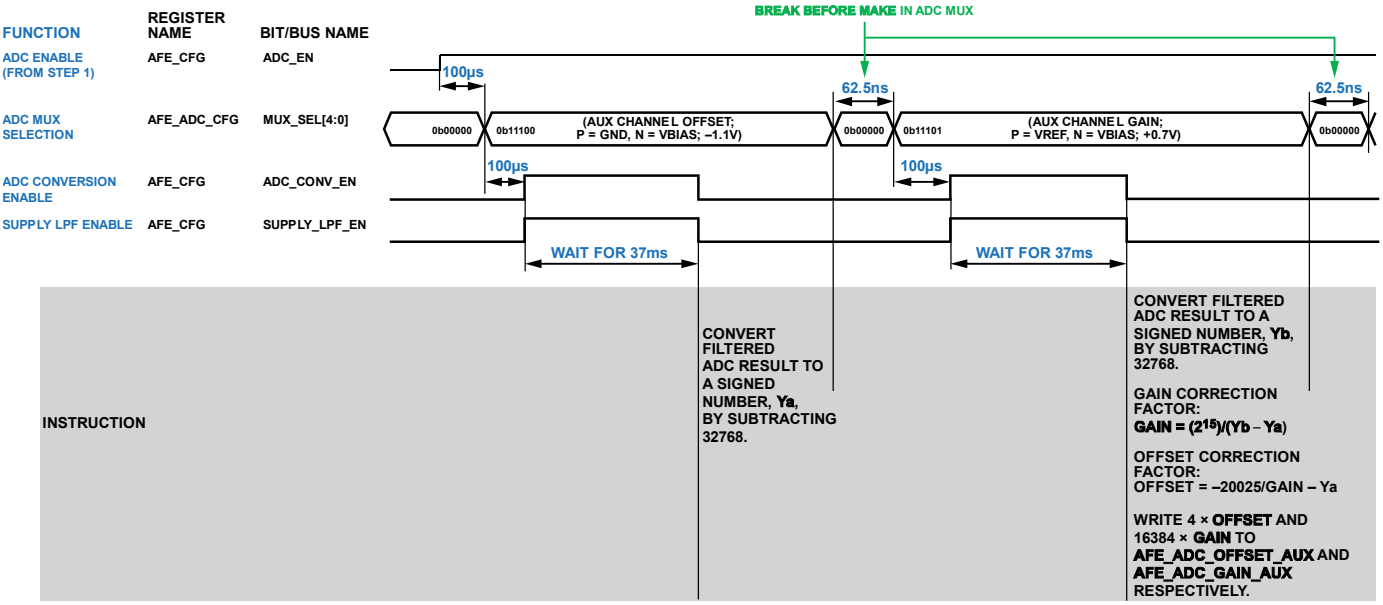


Figure 132. AFE Power-Up

Step 2. Auxiliary Channel Calibration



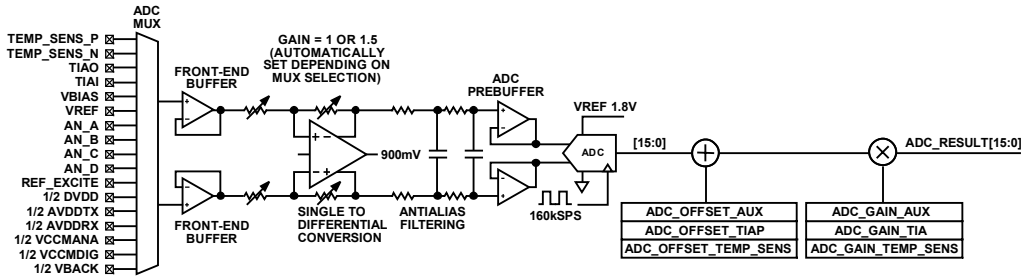


Figure 133. Auxiliary Channel Calibration

Step 3. Auxiliary Channel Measurement

This step is recommended for an auxiliary channel measurement when using a thermistor. A ratiometric measurement is performed with a gated reference voltage measurement, REF_EXCITE, and an auxiliary channel measurement, AIN_A. For a standard auxiliary voltage measurement, use only the AIN_A channel measurement in standalone mode.

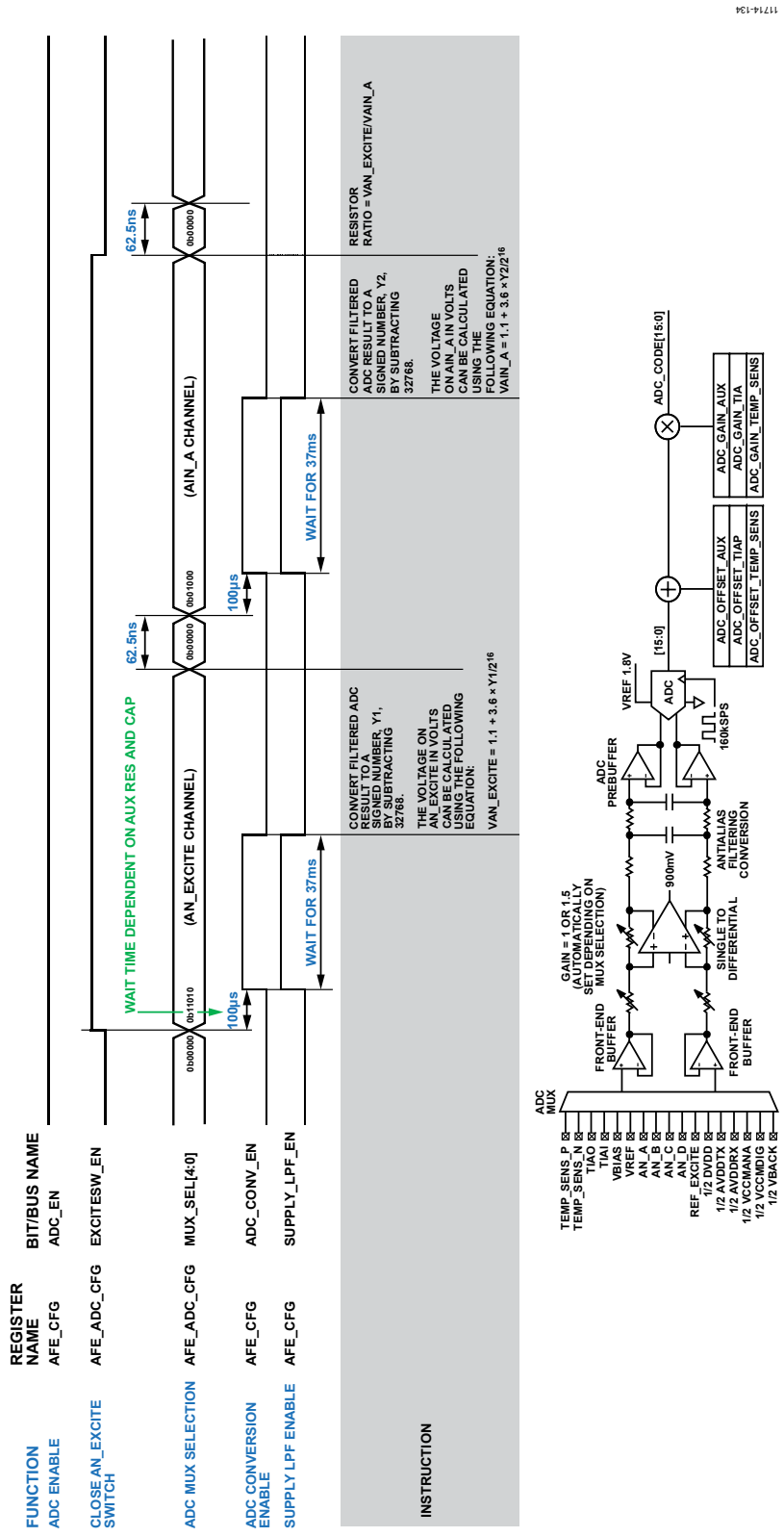


Figure 134. Auxiliary Channel Measurement

Step 4. Internal Temperature Sensor Calibration

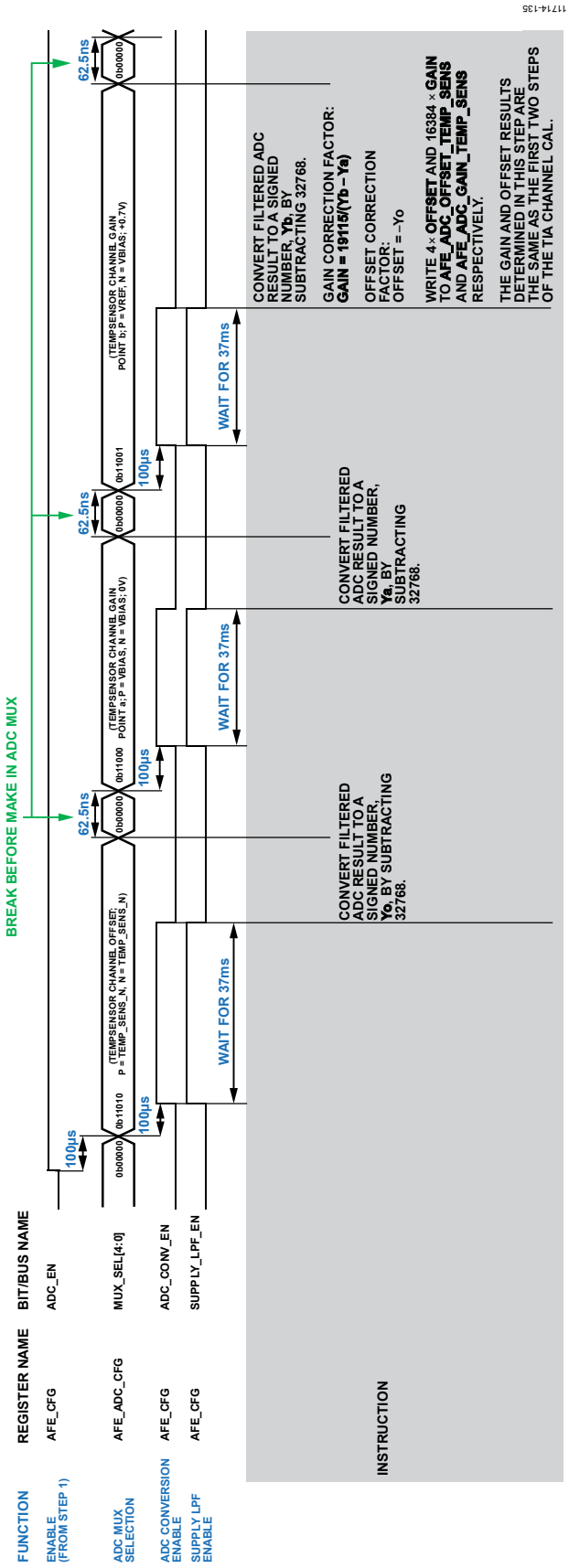


Figure 135. Internal Temperature Sensor Calibration

Step 5. Internal Temperature Sensor Measurement

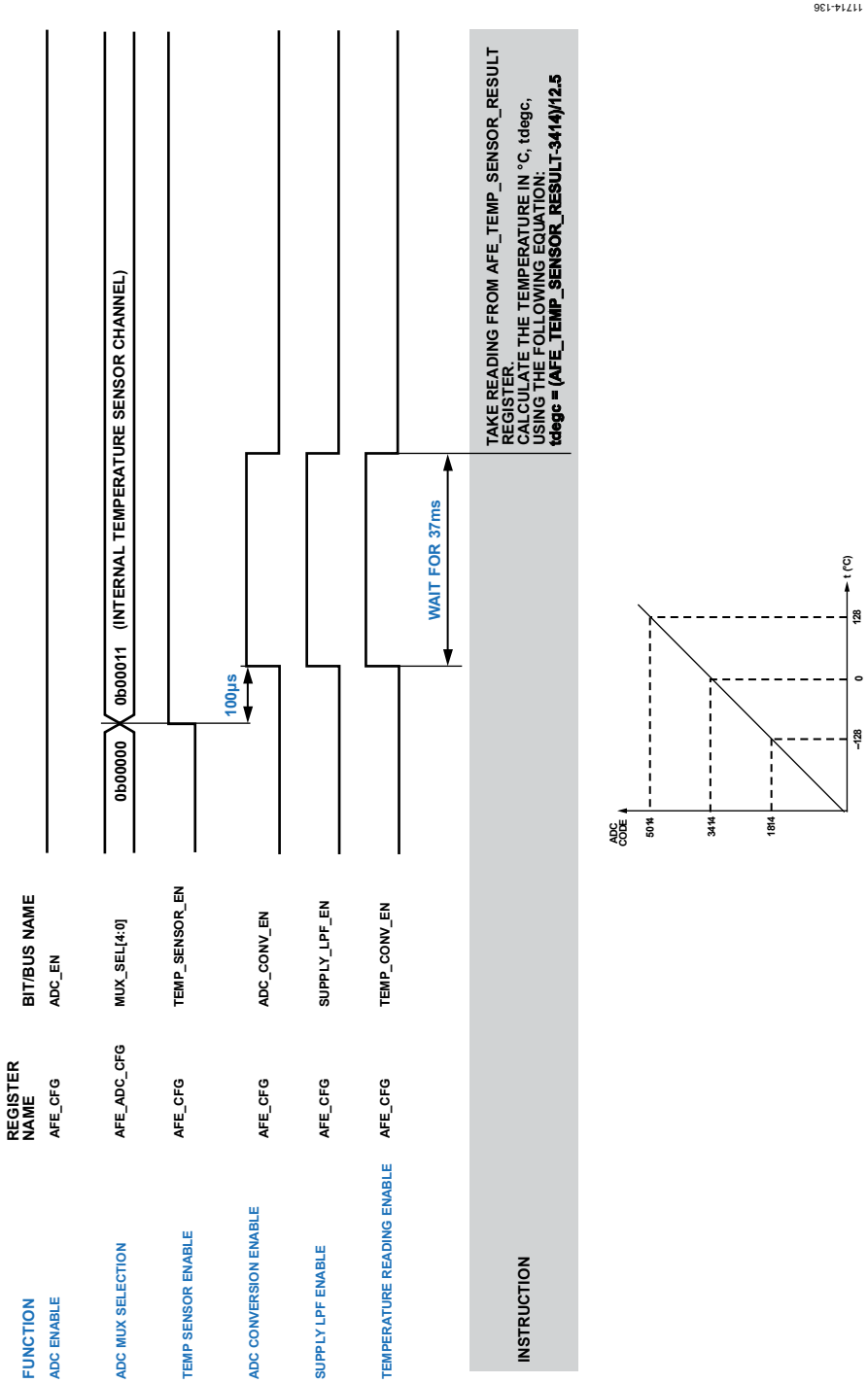


Figure 136. ADC Code vs. Temperature Range

Step 6. Excitation Power-Up Sequence

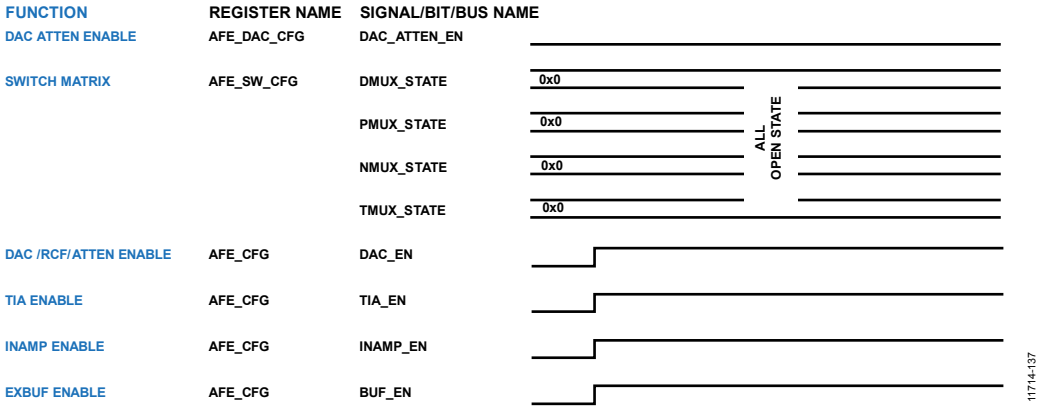


Figure 137. Excitation Power-Up Sequence

Step 7. TIA Channel Calibration

Assumes RCAL = 1 kΩ.

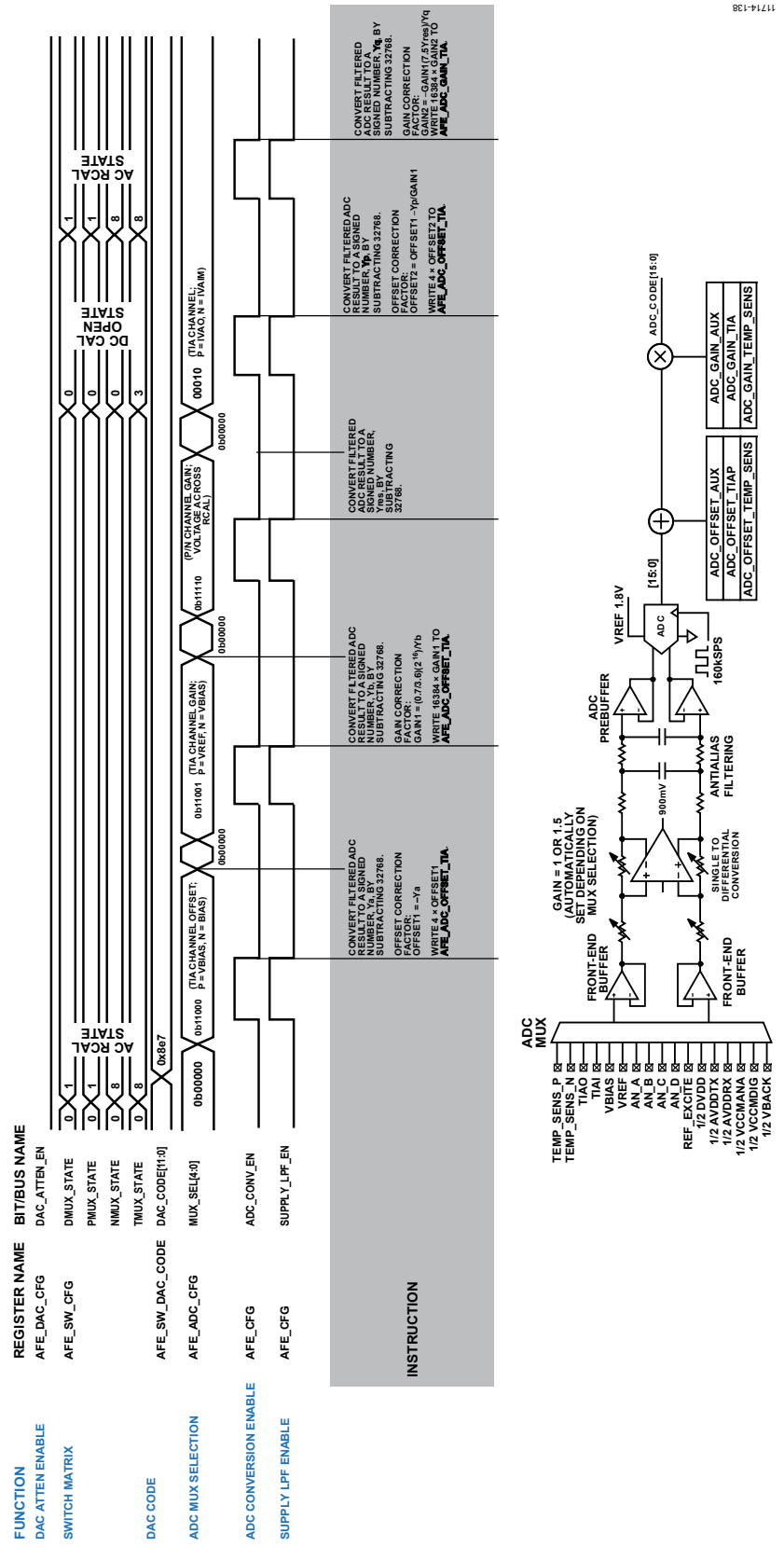


Figure 138. TIA Channel Calibration

Step 8. Excitation Channel Calibration (Attenuator Mode)

Assumes RCAL = 1 kΩ.

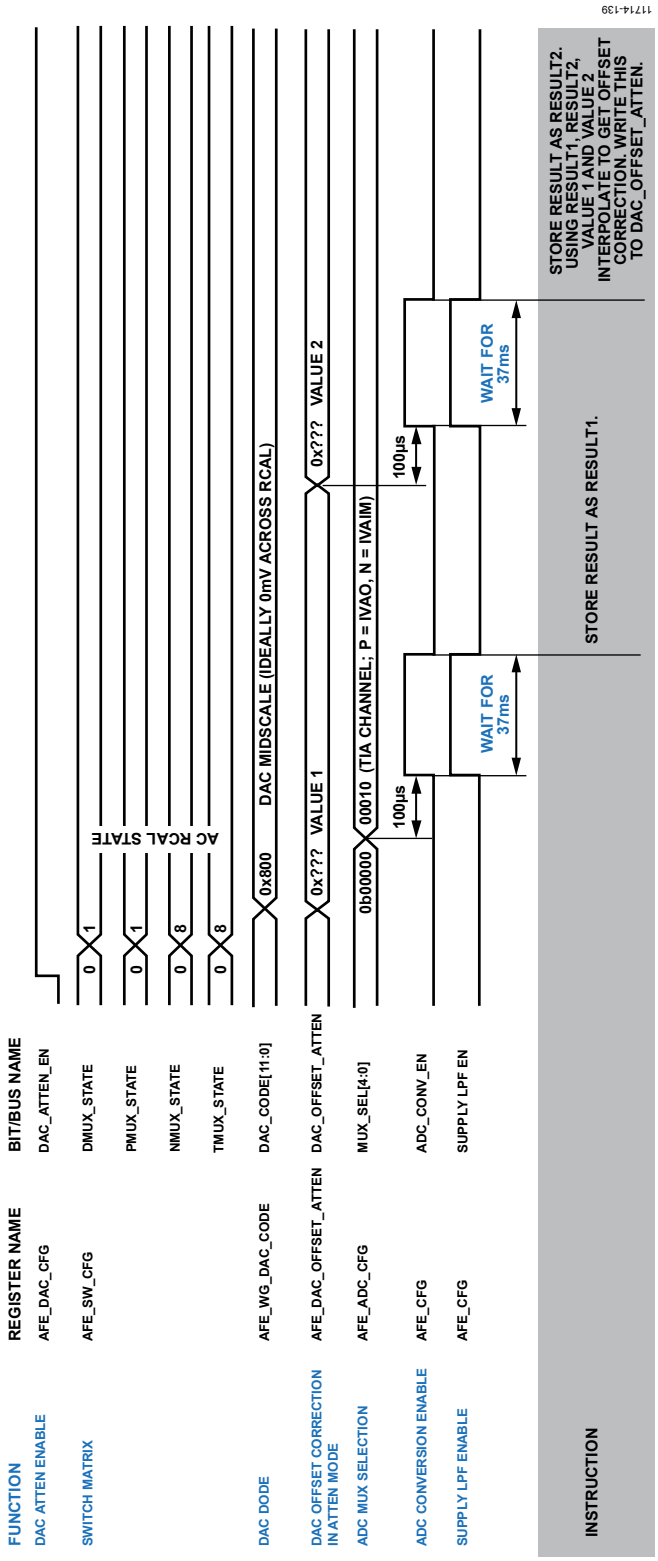


Figure 139. Excitation Channel Calibration (Attenuator Mode)

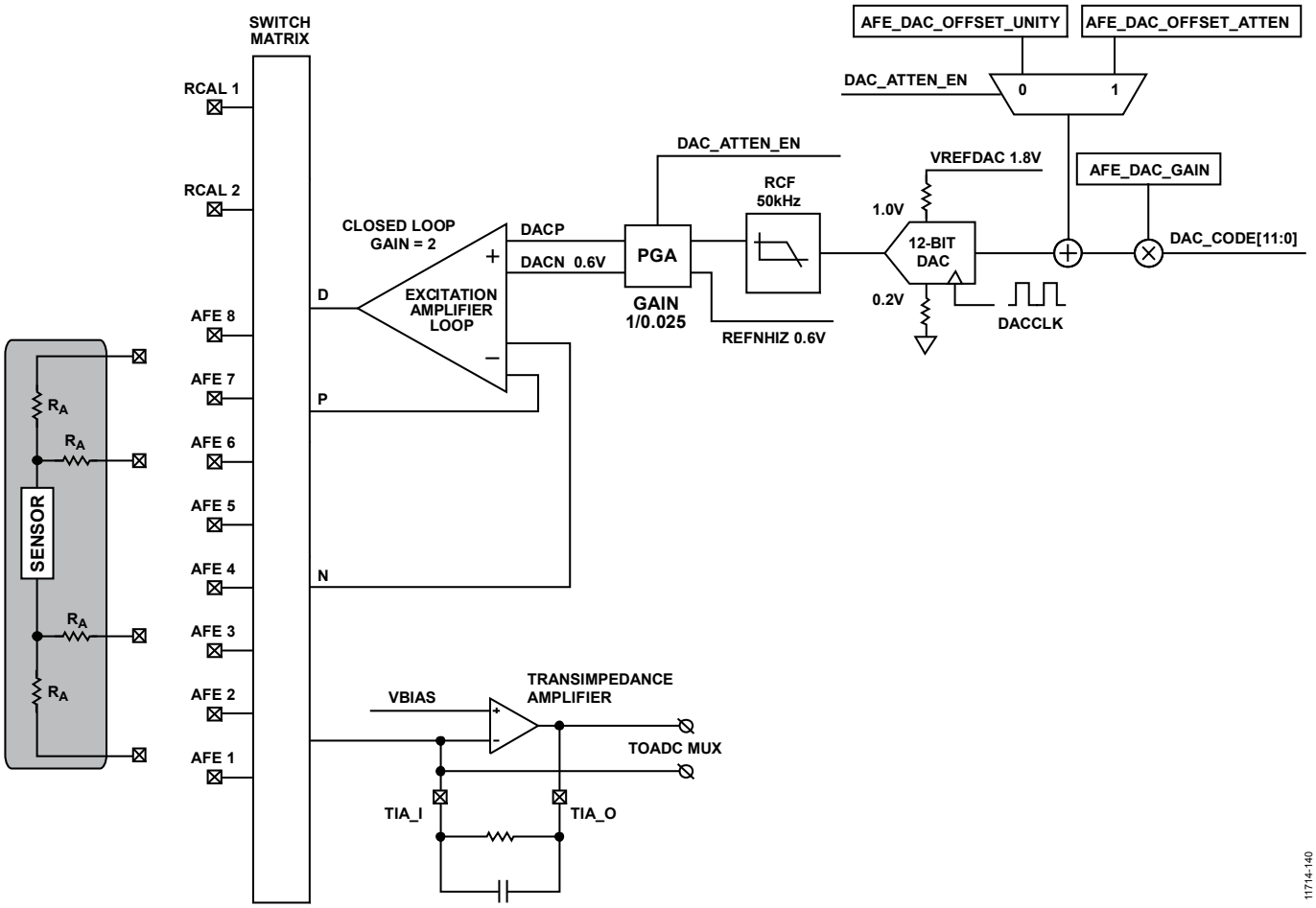


Figure 140. Excitation Channel Calibration (Attenuator Mode)

Step 9. Excitation Channel Calibration (Nonattenuator Mode)

Assumes RCAL = 1 kΩ.

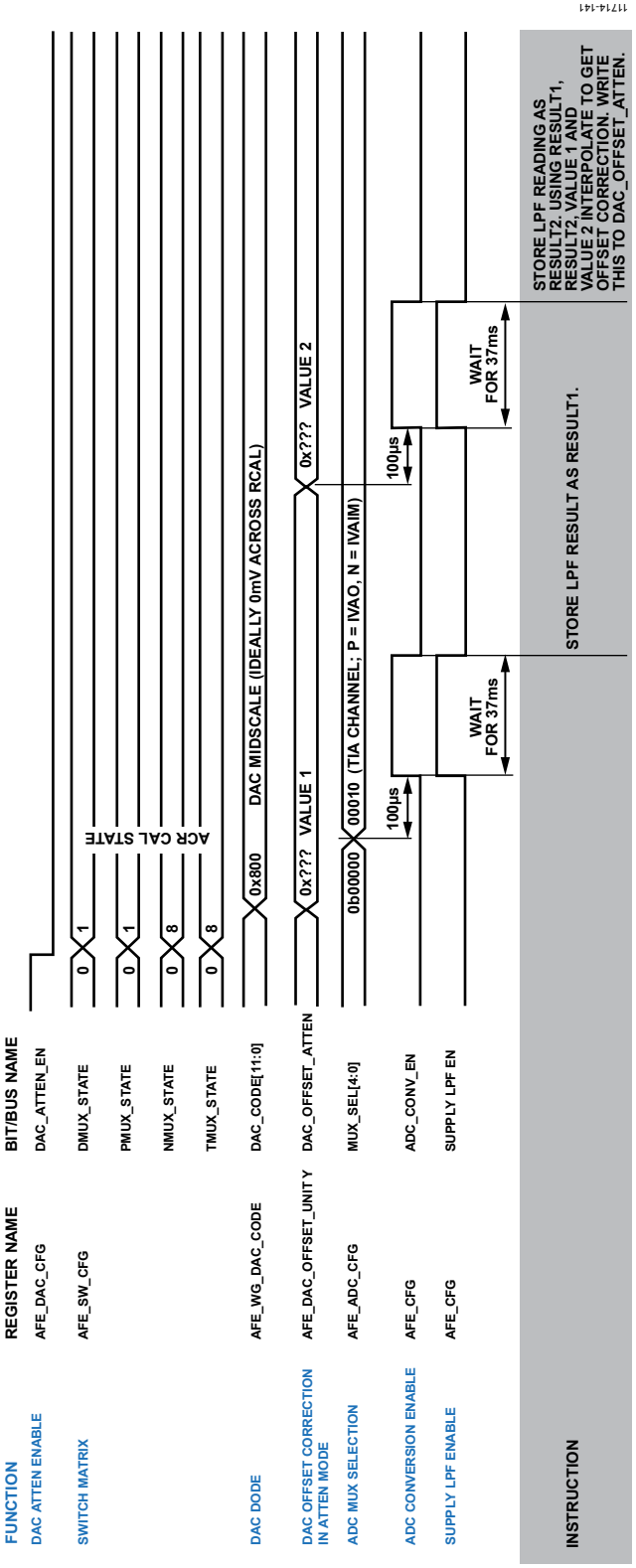


Figure 141. Excitation Channel Calibration (Nonattenuator Mode)

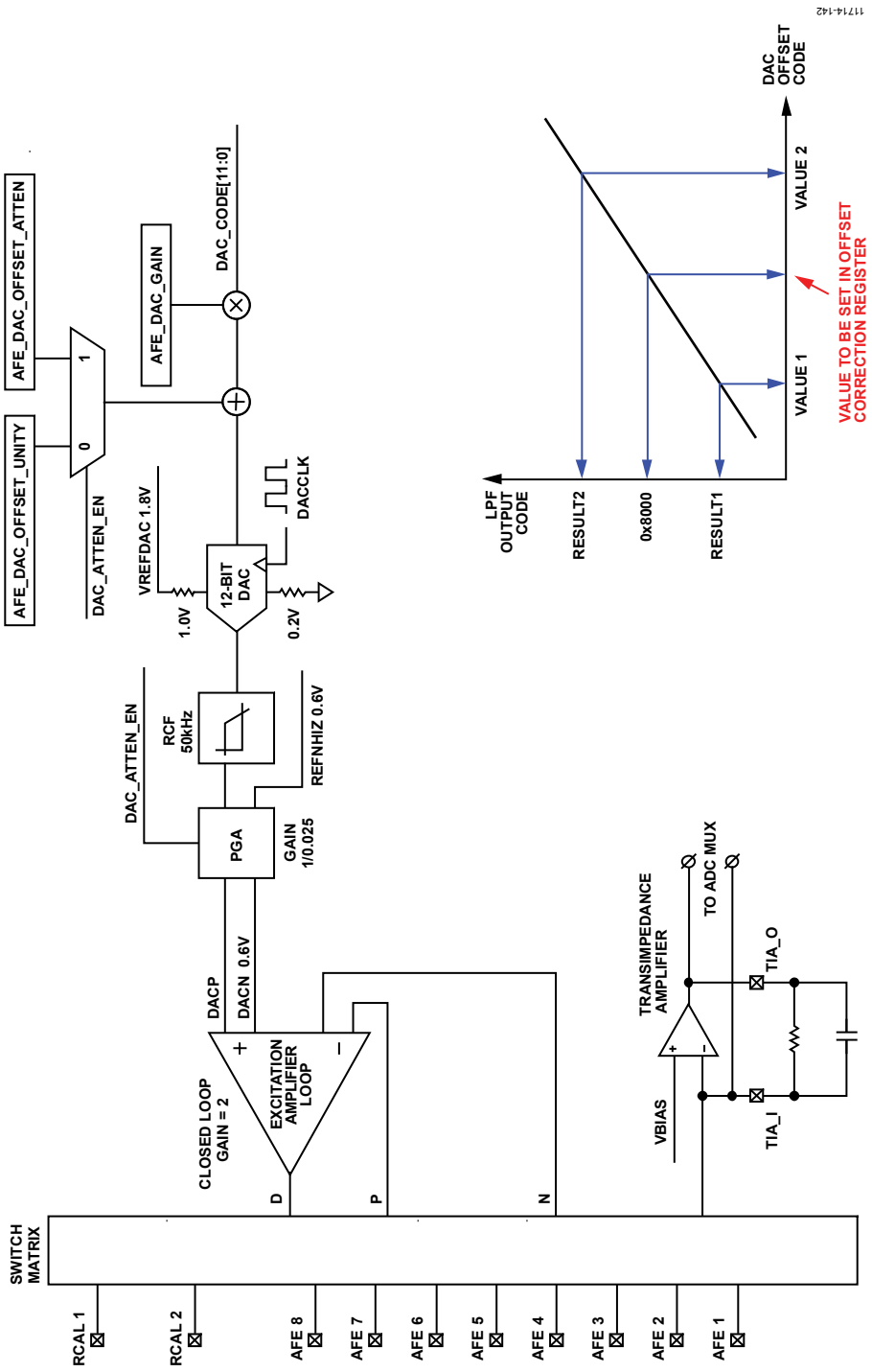


Figure 142. Excitation Channel Calibration (Nonattenuator Mode)

Step 10. AC Measurement Example

Assumes 4-wire switch matrix configuration 10 kHz excitation.

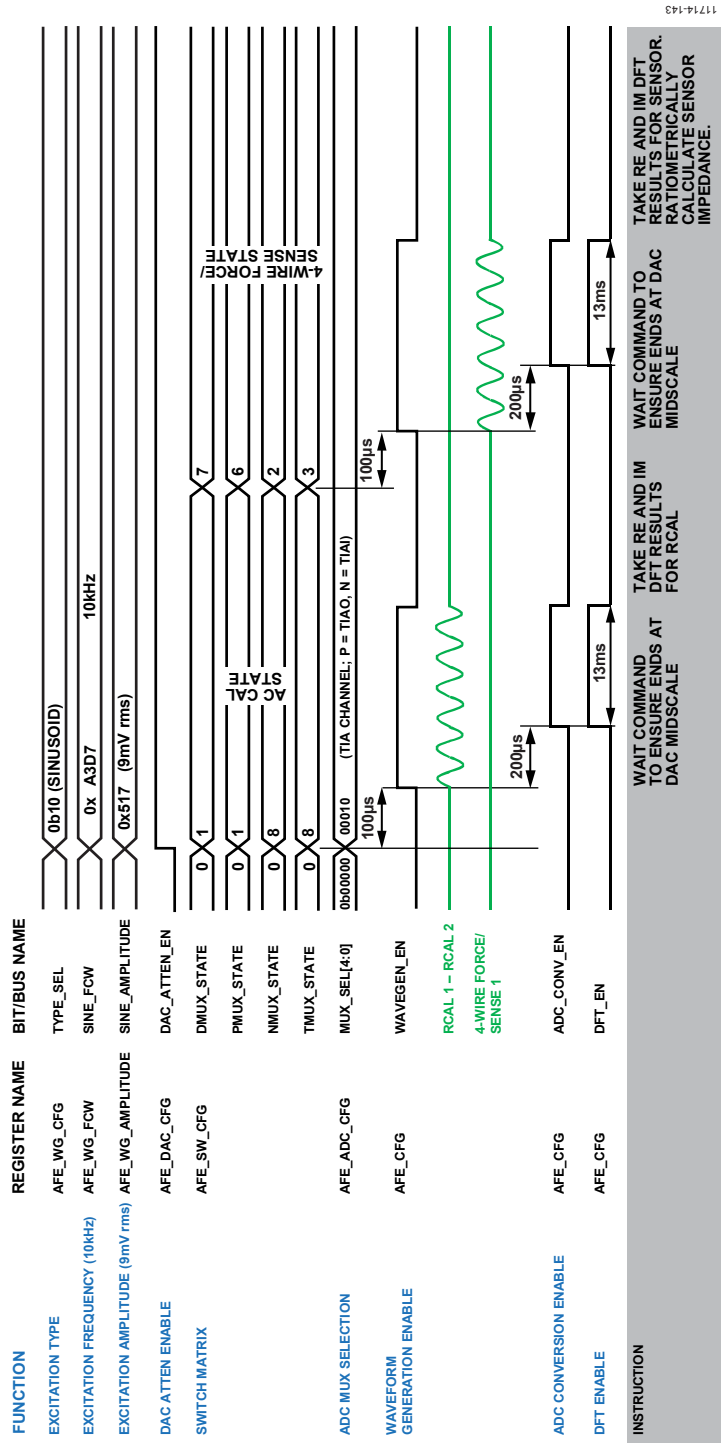


Figure 143. AC Measurement Example

Step 11. Attenuation Change

This step is recommended to minimize the glitch seen by the sensor when ATTEN is changed from H to L. The assumption here is that the switch matrix is in the example 4-wire force/sense state, as it is during sensor excitation. This step is not necessary if up to ~100 mV of a glitch for ~20 µs can be tolerated.

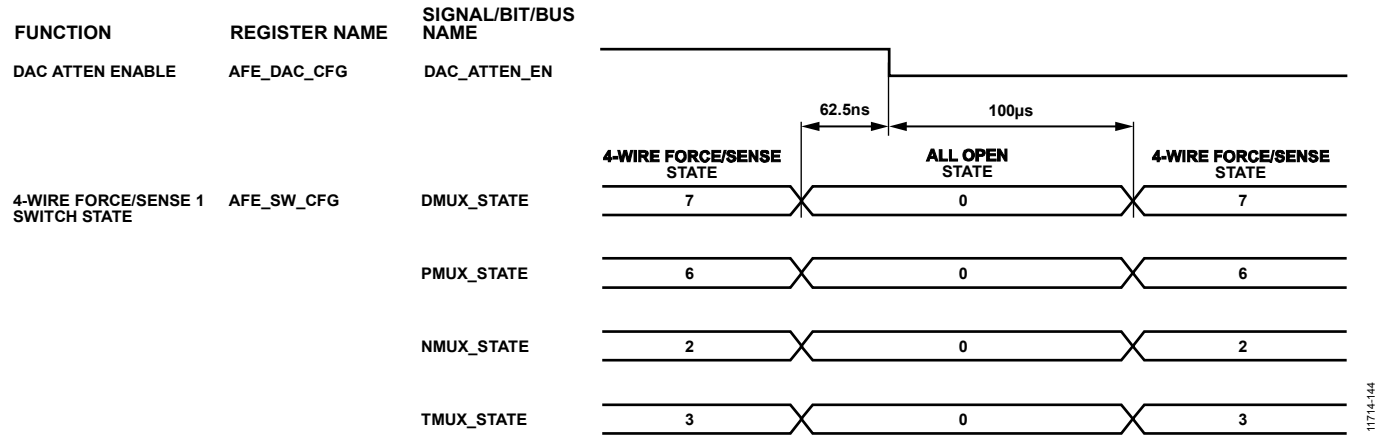


Figure 144. Attenuation Change

Step 12. DC Measurement Example

Assumes 4-wire force/sense switch state. Trapezoidal excitation. DAC CLK is default of 326.530 kHz.

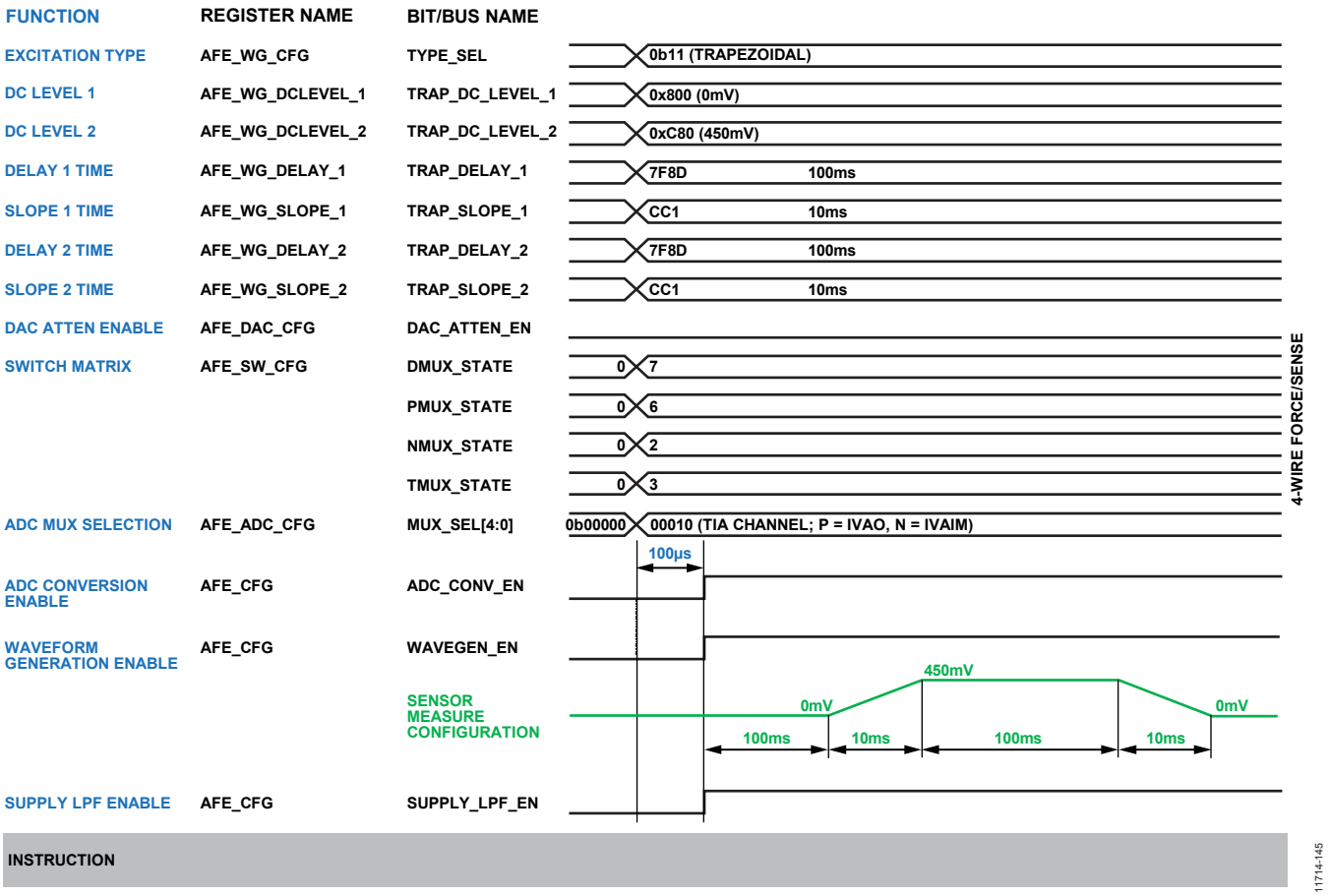


Figure 145. DC Measurement Example

Step 13. AFE Power-Down

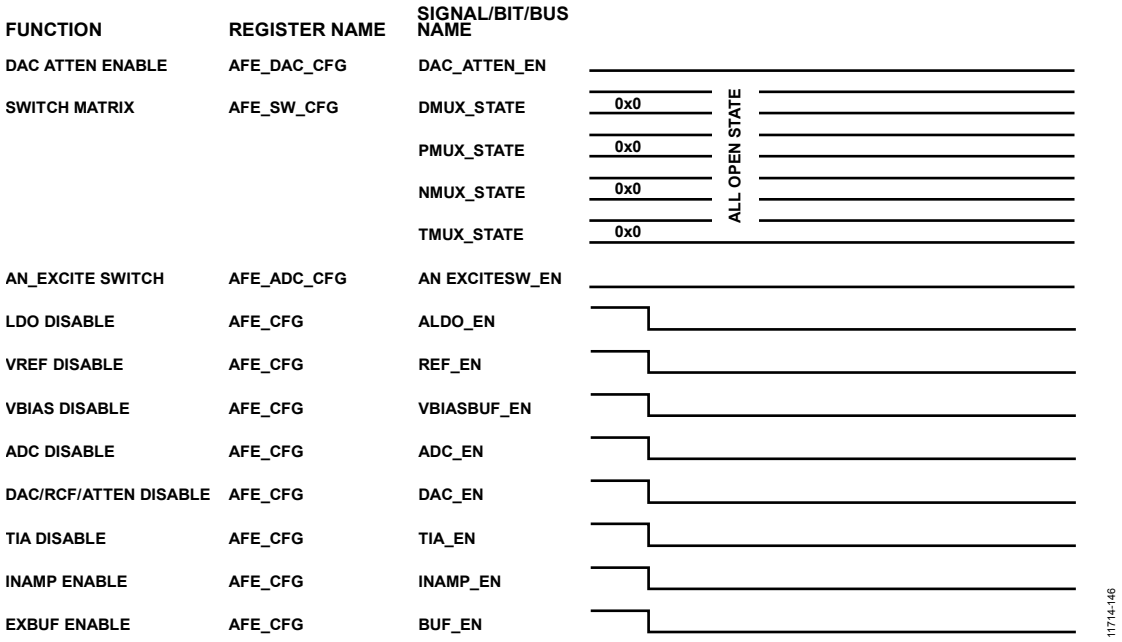


Figure 146. AFE Power-Down

PRECISION VOLTAGE REFERENCE AND BIAS

INTRODUCTION

This section captures the integrated precision voltage reference and bias block for the AFE. This block generates accurate voltage references (VREF and VBIAS) for the ADC and an accurate voltage reference (VREFDAC) for the DAC. It also generates bias currents for various blocks on the AFE. VREF can be used to drive an external resistor or thermistor string to ground through the AN_EXCITE pin. VREF and VBIAS must be decoupled externally using 4.7 μF and 0.47 μF capacitors, respectively. VREFDAC is decoupled internally and does not have a dedicated pin.

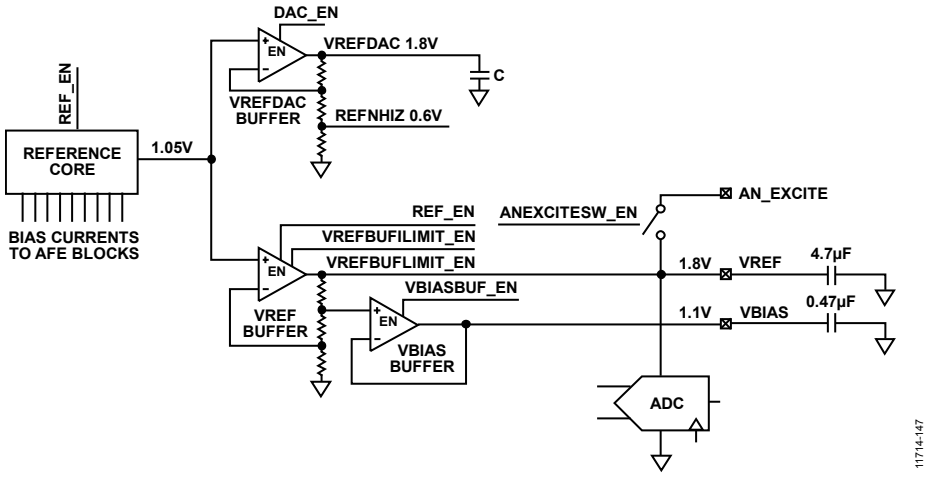


Figure 147. Signal Acquisition Reference Circuit

PRECISION REFERENCE REQUIREMENTS

The precision reference specifications are driven by the overall system accuracy targets, in particular for the dc excitation phase of the external sensor, where the accuracy of the current measurement is important.

Precision Voltage Usage

The reference core is supplied by the AFE regulated supply, AVDD; therefore, the AFE regulator must be enabled before the reference is enabled by setting the ALDO_EN bit in the AFE_CFG register. The VREF buffer, VBIAS buffer, and VREFDAC buffer are supplied by VCCM. The precision reference generates bias currents for the DAC, RCF, attenuator, excitation loop, TIA, and ADC channels; therefore, the reference must first be enabled before any of those blocks are enabled. The reference is enabled by setting the REF_EN bit in the AFE_CFG register. To prevent an excessive dip in the battery voltage, it is recommend to also set the VREFBUFILIMIT_EN bit in the AFE_CFG register to limit the current drawn from VCCM to be less than 2 mA while the 4.7 μF capacitor on VREF is being charged. The maximum charge time for VREF is 20 ms, after which it is recommended to clear the VREFBUFILIMIT_EN bit to minimize the output impedance of the VREF buffer for the maximum ADC performance. Again, to limit the current from VCCM, it is recommended to only enable the VBIAS buffer 20 ms after VREF has been enabled. The VBIAS buffer is enabled by setting the VBIASBUF_EN bit in the AFE configuration register.

The recommended sequence is as follows:

1. Enable the analog LDO (ALDO_EN bit in the AFE_CFG register) and LDO current limit (ALDOILIMIT_EN in the AFE_CFG register).
2. Wait 2 ms.
3. Enable the VREF and VREF current limits (the REF_EN bit and the VREFBUFILIMIT_EN bit in the AFE_CFG register).
4. Disable the LDO current limit.
5. Wait 20 ms.
6. Disable the VREF current limit (VREFBUFILIMIT_EN bit in the AFE_CFG register).
7. Enable the VBIAS buffer (VBIASBUF_EN bit in the AFE_CFG register).
8. Wait 5 ms.

See the Proposed Calibration Routine for Auxiliary Channel section of the No Factory Calibration section for more details.

The DAC buffer that creates the reference voltage, VREFDAC, for the DAC is automatically enabled when the DAC is enabled by setting the DAC_EN bit in the AFE_CFG register.

VREF can also be used to drive an external resistor or thermistor string to ground through the AN_EXCITE pin by closing the AN_EXCITE switch and setting the ANEXCITESW_EN bit in the AFE_ADC_CFG register. The minimum recommended resistive loading to ground is 9 kΩ, resulting in a maximum recommended dc current loading on VREF of 200 μA.

Precision Voltage Reference vs. RCAL

The voltage reference works in combination with the RCAL external reference resistor. The system is limited by a combination of both components, and improved RCAL may be considered by the user to improve performance

DAC AND LOW-PASS RCF

INTRODUCTION

This section describes the signal generating DAC of the sensor that is coupled through a low-pass reconstruction filter (RCF). The RCF filters out the DAC waveform before driving the excitation amplifier, which buffers this signal to drive the sensor or reference resistor. The digital waveform generation subblock is described in the Analog Front-End Interface section.

DAC AND LOW-PASS RCF IN SIGNAL CHAIN

The DAC is driven via the waveform generator as described in the Analog Front-End Interface section. There are multiple digital modes of operation and digital hardware accelerators for signal generation.

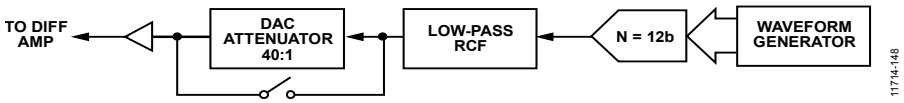


Figure 148. Simplified Signal DAC Chanel Block Diagram

The ADuCM350 fully integrates the low-pass reconstruction filter design on chip, removing the external component cost, variation, and interconnect. The excitation amplifier also has finite small signal bandwidth and serves to further filter the generated signal.

The DAC attenuator block (divide by 40) is enabled for ac excitation measurements and is disabled for dc excitation measurements by setting the DAC_ATTEN_EN (Bit 0) of the AFE_DAC_CFG register to 0 (also the reset condition). The DAC offset in attenuator mode is set by the AFE_DAC_OFFSET_ATTEN register.

The DAC offset in unity gain register (AFE_DAC_OFFSET_UNITY) and the DAC gain register (AFE_DAC_GAIN) are available to ensure the 600 mV peak-to-peak maximum output swing from the DAC. For more details, refer to the No Factory Calibration section.

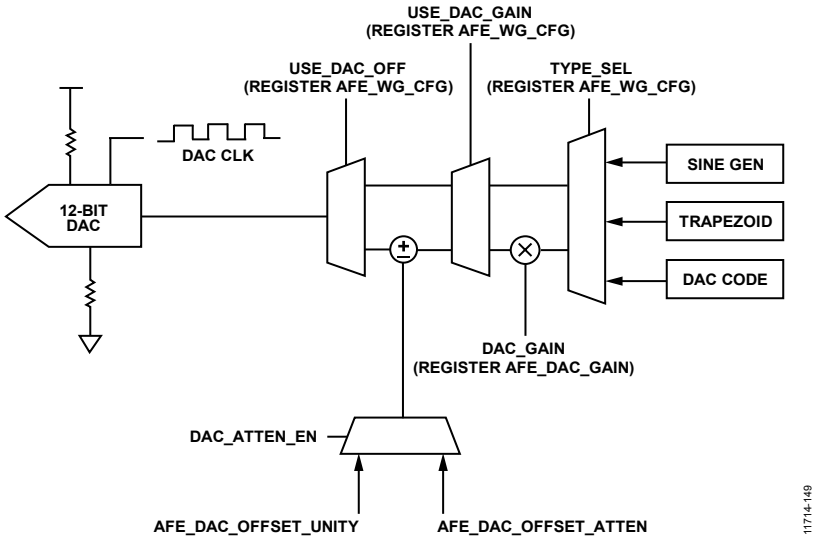


Figure 149. DAC Signal Control

DAC Digital

The DAC digital functionality is enabled when the DVDD power supply is powered up. The analog circuitry in the DAC, RCF, and PGA is powered up with the DAC_EN bit (Bit 6) of the AFE_CFG register. The power-on reset state of operation for DAC, RCF, and PGA analog circuitry is power-down mode.

The positive reference for the DAC is coupled to the precision voltage reference on the ADuCM350; therefore, the precision reference must also be enabled with the DAC in operation.

The DAC digital datapath is described in more detail in the Analog Front-End Interface section.

The digital content and control of the DAC attenuator is described in the DAC Attenuator section.

DAC ATTENUATOR

The DAC signal can be attenuated if desired by switching in a voltage to voltage attenuator via setting the DAC_ATTEN_EN bit (Bit 0) of the AFE_DAC_CFG register. Enabling this bit attenuates the signal by 1/40.

The magnitude of the output signal can be varied by both the DAC word scalar in the waveform generation and the attenuator in the analog lineup. For minimum glitch on the sensor, it is recommended to change the DAC_ATTEN_EN bit only when switch states are all open. The recommendation of a wait of 100 μs before changing the switch matrix back to the sensor measurement state is derived from 5 × 20 μs, where 20 μs is an RC time constant of a single pole at 50 kHz.

Table 552. DAC ATTEN_EN Bit

DAC_ATTEN_EN	Typical
0	Gain of −1
1	Gain of −0.025

The operational amplifiers in the signal chain also provide additional filtering of the DAC signal.

This block is powered down when not enabled via the DAC_ATTEN_EN bit in the AFE_DAC_CFG register to minimize power consumption.

The inversion in the PGA stage, as shown in Figure 149, is canceled by a second inversion in the signal generation chain such that the low digital level for the DAC corresponds to the low analog level at the load.

DAC SIGNAL GENERATION REQUIREMENTS

The low-pass RCF requirements are set by the full system requirements from two perspectives:

- Quality of the sensor excitation signal; repeatability is key.
- ADC aliasing requirements; minimize errors.

For signal quality, high quality filtering of the signal maximizes the repeatability of the signal across the sensor, which is a key motivator for good ac response.

A 12-bit string DAC is used to generate the required voltage. The DAC clocking frequency is 4-bit programmable. The theoretical full signal swing from the DAC, shown in Figure 150, is a ±400 mV peak-to-peak swing centered at 600 mV. Assuming there are perfect amplifiers, this translates to a range of 1.6 V (±800 mV peak-to-peak). The amplifiers support a maximum voltage swing of ±600 mV at RCAL or sensor. The excess range on the DAC core is a conservative excess to allow for power-up offset correction (in both attenuation modes) and Analog Devices factory gain correction.

Amplifier A1 and Amplifier A2 buffer the signal and set up a differential output from the DAC. Amplifier A3 allows software controlled for gain attenuation and a first-order 50 kHz reconstruction filter. A follow on inversion stage is carried out in the AFE excitation loop stage so that the sensor sees a positive voltage swing.

The AFE excitation loop sets the voltage across the sensor to be equal to twice the differential DAC output voltage. As the force and sense points differ, the AFE excitation loop provides an accurate method of controlling the voltage across the sensor through the access resistances.

PGA + DAC Output Range = −400 mV to +400 mV

PGA + DAC LSB Size = 0.8 V/4095 = 195.36 μV

where the output range, and therefore LSB size, is gained up by 2 with excitation amplifier gain (see the AFE Excitation Loop section for more information about transmit stage signal swings).

See the Analog Front-End Interface section for an example calculation for setting trapezoidal output at sensor/RCAL.

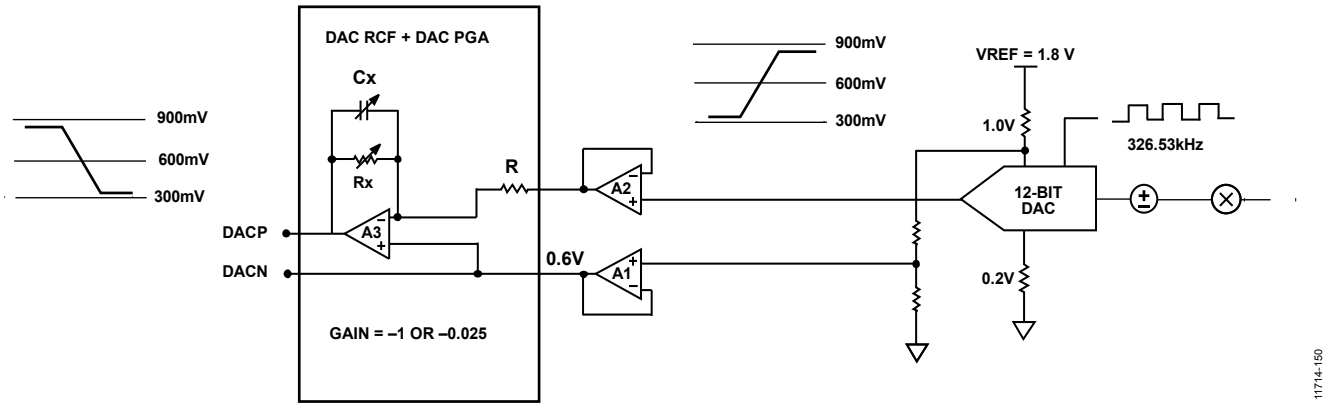


Figure 150. DAC and RCF Signal Chain Block Diagram (Nominal Usage Case with 600 mV Swing)

LP RCF OPERATION

The low-pass RCF with 3 dB corner frequency at 50 kHz serves to smooth/filter the DAC transitions. The 3 dB corner frequency is set during the Analog Devices factory calibration. The trim code for RCF is loaded when the kernel is loaded at boot.

ADC CHANNEL

INTRODUCTION

In this section, an overview of the ADC channel for the AFE is presented. The ADC is the core of the precision measurement engine of the ADuCM350 and is used for sensor measurement, temperature measurement, on-chip supply voltage measurement, and auxiliary analog input voltage measurement.

See Figure 151 for a block diagram representation of the ADC channel. The channel consists of a channel selection multiplexer, front-end buffering, single-to-differential conversion, antialias filtering, ADC prebuffering, ADC core, and digital filtering and decimation.

ADC CHANNEL BLOCK DIAGRAM

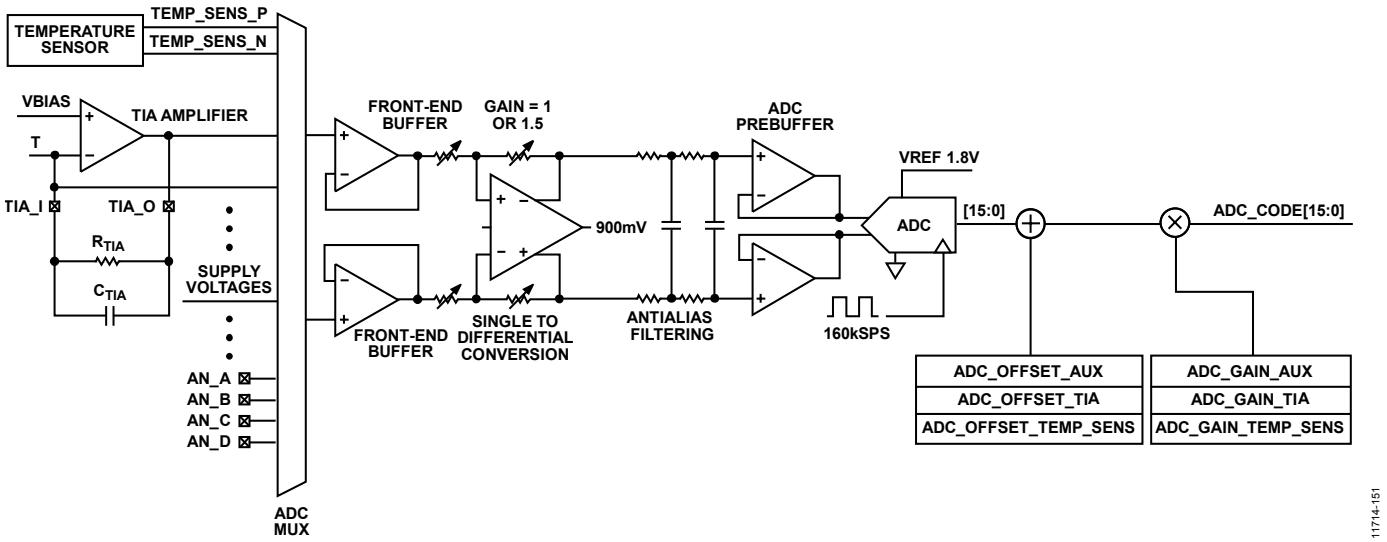


Figure 151. ADC Block Diagram

A description of the function of each block within the ADC channel is as follows:

- ADC multiplexer: selects the desired channel by setting MUX_SEL in the AFE_ADC_CFG register. More detail on the ADC multiplexer is provided in Table 553.
- Front-end buffer: low offset high impedance input buffering of the selected channel for maximum accuracy. This is particularly critical for auxiliary channel measurement of the resistor strings on the auxiliary channels where input current causes an error in the voltage to be measured.
- Optimized for an input voltage range of 0.35 V to 1.8 V, which is consistent with the output range of TIA.
- Single to differential conversion: balances signal to be differential on a common mode of 900 mV so that the signal range is compatible with the ADC core that is referenced to the 1.8 V precision reference.
- Antialias filter: limits the bandwidth of the input signal to suppress aliasing of noise in the conversation. It is implemented as a passive filter with two poles at 54 kHz and 218 kHz.
- ADC prebuffer: buffers signal to allow sampling by the ADC core without distortion.
- ADC core: analog-to-digital conversion at 160 kSPS to a 16-bit data output stream.
- It is important to note that the entire ADC channel is linear to the 16-bit level to guarantee 16-bit performance for measurement of the TIA channel.

In the ac measurement phase, the 16-bit data stream at 160 kSPS is passed through a DFT engine, where the real and imaginary components are calculated to derive that complex impedance of the sensor at the excitation frequency.

In the dc measurement phase, the ADC samples are passed into a digital filter lineup, where the data rate is decimated by 178 SPS to ~900 SPS (898.876 SPS). The data is then passed through a power line reject filter, where there is a minimum of 60 dB attenuation at 50 Hz and 60 Hz.

More information on the DFT engine, digital decimation, and filtering can be found in the Analog Front-End Interface section.

REFERENCE DEPENDENCE

To use the ADC channel, it is critical that the precision voltage reference is enabled. See the Precision Voltage Reference and Bias section for a detailed description of how to power up the reference in a recommended manner.

ADC MULTIPLEXER

The available channels to the ADC multiplexer are as listed in Table 553. Note that because the ADC channel is differential, each channel has a positive and negative side. The differential measurement gives robustness and repeatability to the accuracy of measurement in the presence of any common-mode noise, in particular for transimpedance measurements and temperature sensor measurements. For this reason, the auxiliary channels and supply channels are all measured with respect to VBIAS, 1.1 V, which is a compatible reference point for the front-end buffers.

The selection of the desired channel is done by setting the MUX_SEL bits in the AFE_ADC_CFG register to the correct value. The gain of each channel is automatically set when setting the respective MUX_SEL bit, whether it is 1.0 or 1.5. It is recommended when changing channels to first set MUX_SEL to 00000 for at least one master clock cycle (62.5 ns) before setting these bits to the desired channel to give a break-before-make operation. It is recommended that a 100 µs delay be used when changing the ADC multiplexer channel to allow the voltage through the antialias filter to settle because it has a group delay of tens of microseconds.

Table 553. ADC Input Multiplexer

Signal	MUX_SEL[4:0]	P	N	Gain	Purpose
Floating	00000				No input. Used for break-before-make when changing channel. Recommendation is to go to this state for at least one master clock cycle (62.5 ns) while changing channel.
TIA	00010	IVAO	IVAIM	1.5	TIA output. This is a differential measurement directly across the TIA resistor. This gives optimum accuracy because any offset and noise from the TIA op amp does not contribute an error source on the TIA measurement.
Temperature Sensor	00011	TEMP_SENS_P	TEMP_SENS_N	1.5	Internal temperature sensor channel.
Auxiliary and Supply	01000	AN_A	VBIAS	1	Measure AN_A with respect to VBIAS (1.1 V).
	01001	AN_B	VBIAS	1	Measure AN_B with respect to VBIAS (1.1 V).
	01010	AN_C	VBIAS	1	Measure AN_C with respect to VBIAS (1.1 V).
	01011	AN_D	VBIAS	1	Measure AN_D with respect to VBIAS (1.1 V).
	01100	REF_EXCITE	VBIAS	1	Measure REF_EXCITE with respect to VBIAS (1.1 V).
	01101	½DVDD	VBIAS	1	Measure ½ DVDD with respect to VBIAS (1.1 V).
	01110	½AVDDTX	VBIAS	1	Measure ½ AVDDTX with respect to VBIAS (1.1 V).
	01111	½AVDDRX	VBIAS	1	Measure ½ AVDDRX with respect to VBIAS (1.1 V).
	10000	½VCCMANA	VBIAS	1	Measure ½ VCCMANA with respect to VBIAS (1.1 V).
	10001	½VCCMDIG	VBIAS	1	Measure ½ VCCMDIG with respect to VBIAS (1.1 V).
	10010	½VBACK	VBIAS	1	Measure ½ VBACK with respect to VBIAS (1.1 V).
Calibration	11000	VBIAS	VBIAS	1.5	Measure TIA ADC channel offset for calibration.
	11001	VREF	VBIAS	1.5	Measure TIA and temperature sensor ADC channel gain error for calibration.
	11010	TEMP_SENS_N	TEMP_SENS_N	1.5	Measure temperature sensor channel offset for calibration.
	11100	0	VBIAS	1	Measure auxiliary/supply channel offset for calibration.
	11101	VREF	VBIAS	1	Measure auxiliary/supply channel gain error for calibration.
	11110	P	N	1.5	Measure P minus N voltage for TIA and ADC channel calibration. See the No Factory Calibration section for use of this channel.
	Other codes				Reserved.

ADC CALIBRATION

The gain and offset errors for the temperature sensor, auxiliary channels, and external sensor are incorporated in the respective gain and offset registers, AFE_ADC_GAIN_X and AFE_ADC_OFFSET_X. The respective gain and offset register can be selected using the GAIN_OFFS_SEL bits in the AFE_ADC_CFG register. Refer to the Analog Front-End Interface section for details. Note that these registers are not retained when the device is in hibernate mode and must be set as 1 (AFE_ADC_GAIN_X) and 0 (AFE_ADC_OFFSET_X) in the autocalibration routine.

By default, the gain on the output of the ADC is 1, and the offset is 0. The calibration routine is covered in more detail in the No Factory Calibration section.

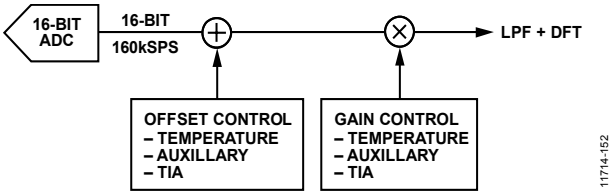


Figure 152. ADC Gain and Offsets for Calibration

ADC MEASUREMENT

The receive stage consists of a channel selection multiplexer, front-end buffering, single to differential conversion, antialias filtering, ADC prebuffering, ADC core, and digital filtering and decimation. The ADC multiplexer selects the desired channel. Front-end buffers are used to increase accuracy of measurements, particularly for auxiliary channel measurements of resistor strings. The ADC channel is optimized for an input voltage range of 0.35 V to 1.85 V, which is consistent with the output range of TIA. A single to differential conversion balances the signal to be differential on a common mode of 900 mV so that the signal range is compatible with 1.8 V precision reference. Antialias filtering, which consists of a passive low-pass filter with two poles at 54 kHz and 218 kHz, limits the bandwidth of input signal. The resultant signal is prebuffered, and the ADC converts at 160 kSPS to a 16-bit data output stream.

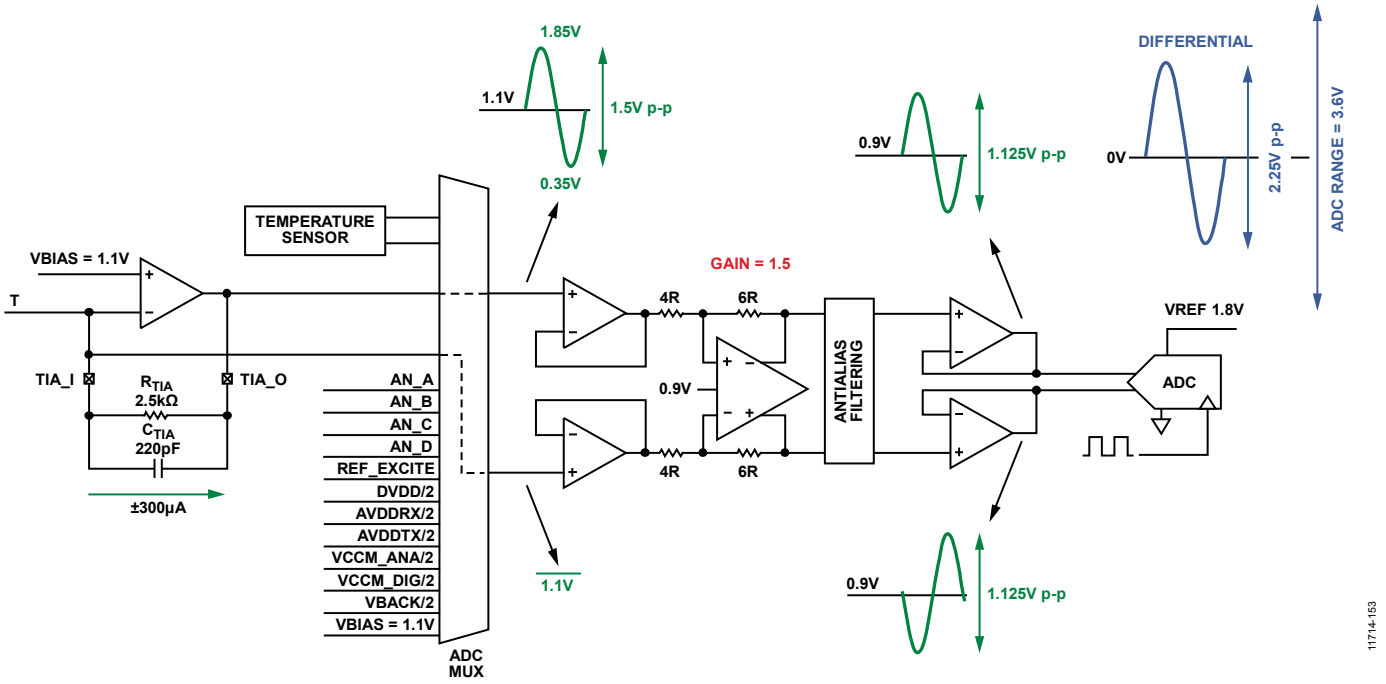


Figure 153. ADC Channel During Sensor Measurement

In Figure 153,

- Gain stage rebalances signal to be balanced differentially.
- TIA_O is sensed with respect to TIA_I; therefore, TIA op amp offset drift and noise are rejected.
- Input Current Resolution = $(3.6\ V/2^{16}) \times (1/1.5) \times (1/R_{TIA}) = 14.6$ nA, where $R_{TIA} = 2.5$ kΩ, as in the example in Figure 153.

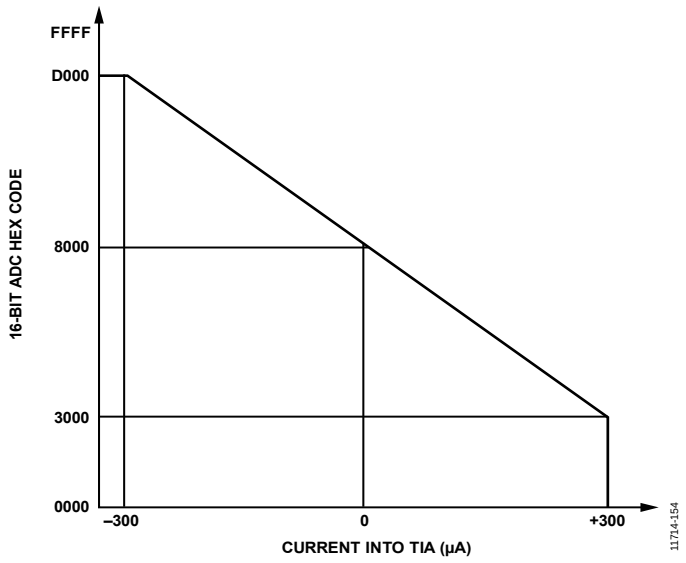


Figure 154. Ideal ADC Channel Transfer Function During Sensor/TIA Measurement

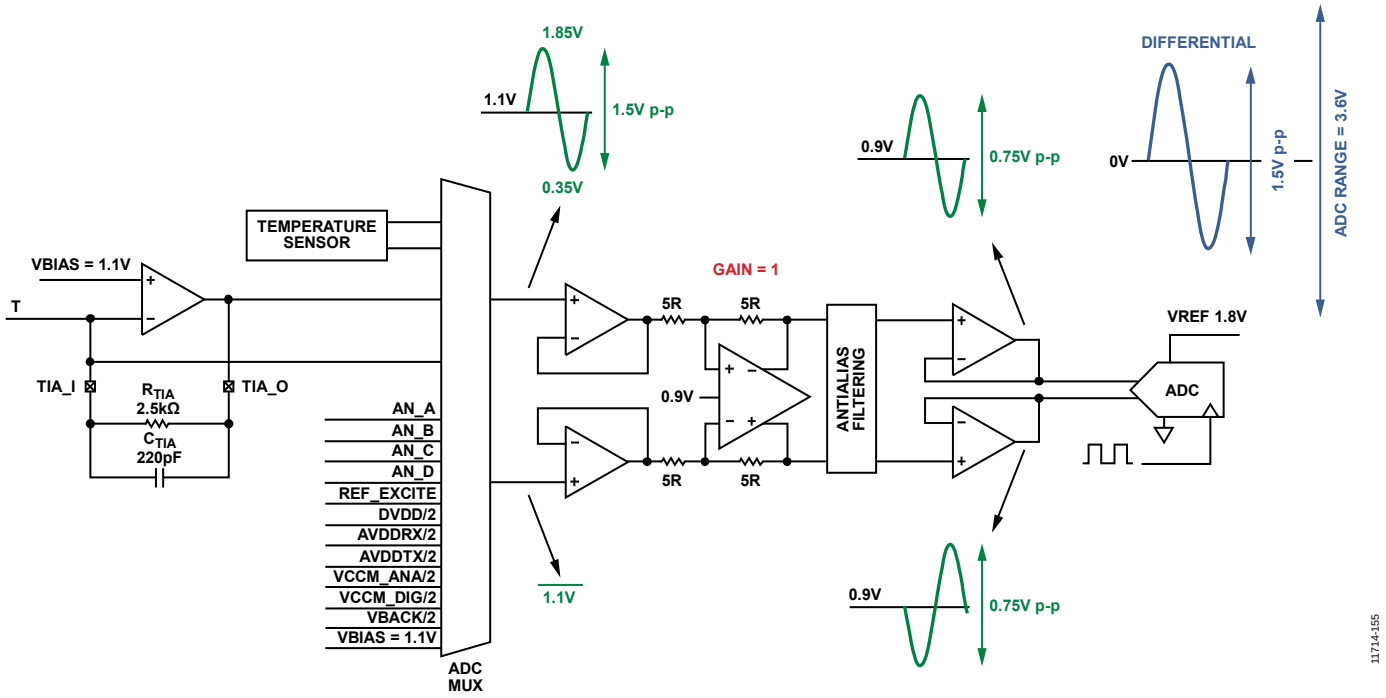


Figure 155. ADC Channel During Auxiliary REF_EXCITE, Supply Channel Measurement

In Figure 155,

- Offset calibration with respect to 0 V on P input and 1.1 V on N input.
- AN_A, AN_B, AN_C, AN_D and REF_EXCITE are sensed with respect to VBIAS (1.1 V).
- Input Voltage Resolution = 3.6 V/2^{16} = 54.9 µV.

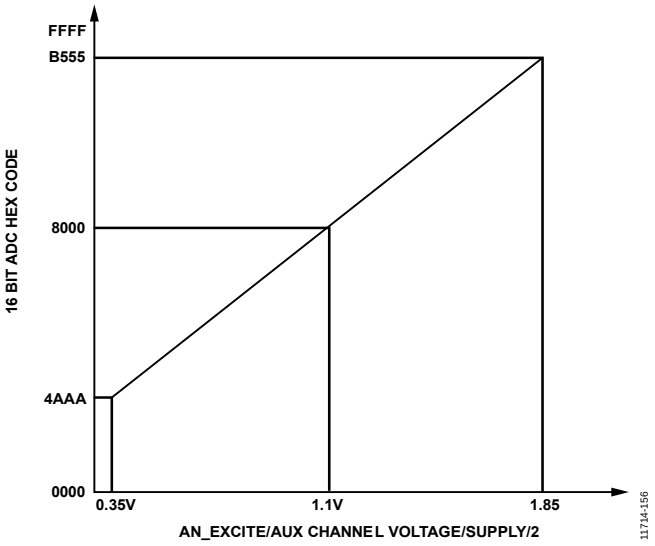


Figure 156. Ideal ADC Channel Transfer Function For Auxiliary, REF_EXCITE, Supply Channel

ADC REGISTERS

ADC

In the AFE Memory Mapped Registers section, there are registers for configuring (AFE_ADC_CFG, AFE_ADC_GAIN_X, AFE_ADC_OFFSET_X) and starting the ADC (AFE_CFG). The ADC executes continuous conversions, in which there is a stream of words on ADCOUT until the user disables the ADC. The raw ADC result can be read from the ADC_AFE_RESULT register.

The range of values of the gain and offset words are described in the Analog Front-End Interface section and are designed to cover the expected variation of offset and gain within the ADC lineup, that is, including the 0.1% tolerance on the 2.5 kΩ resistor.

The output data rate of the ADC depends on which point in the chain is being interrogated. There are two points at which the data can be read:

- 160 kSPS at the output of the gain correction stage.
- ~900 SPS (898.876 SPS) at the output of the power line reject filter.

More information about the DFT engine, decimation, and 50 Hz/60 Hz filtering is available in the Analog Front-End Interface section. The PSRR of the ADC lineup is excellent due to the use of on-chip LDOs to supply power to the ADC as well as the rejection inherent in the lineup.

REF_EXCITE Functionality

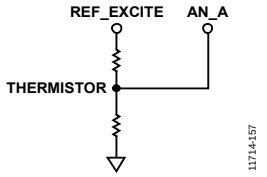


Figure 157. REF_EXCITE Functionality

The REF_EXCITE pin is a gated version of VREF and is controlled by setting the ANEXCITESW_EN bit in the AFE_ADC_CFG register. The REF_EXCITE pin can be used to supply the reference voltage for an external thermistor. The minimum resistance load from REF_EXCITE to ground is 9 kΩ. An uncommitted input, AN_X pin, can be used to measure the divided voltage through the switch mux matrix.

A sample usage case for use of the REF_EXCITE is as follows. Note that AFE LDO, reference, and ADC channel are already enabled and that the ADC auxiliary channel is already gain and offset calibrated, as outlined in the No Factory Calibration section.

1. Close the REF_EXCITE switch by setting the ANEXCITESW_EN bit in the AFE_ADC_CFG register to 1.
2. Wait for the REF_EXCITE switch to power the connected resistor string. The time to wait depends on the size of the resistor and the parasitic capacitance. It is expected that 100 µs is a sufficient wait time with margin.
3. Set the ADC input mux to the REF_EXCITE channel by setting the MUX_SEL bits in the AFE_ADC_CFG to 01100.
4. Wait 100 µs.
5. Enable ADC conversion and 50 Hz/60 Hz rejection filters by setting the ADC_CONV_EN bit to 1 and the SUPPLY_LPF_EN bit to 1.

TEMPERATURE SENSOR MEASUREMENT

INTRODUCTION

This section discusses the integrated temperature sensor functionality of the ADuCM350, using a band gap–based temperature sensor digitized by the on-chip ADC.

The key specification of the temperature sensor is ±1°C typical over the normal usage range.

TEMPERATURE SENSOR ARCHITECTURE AND SPECIFICATIONS

The on-chip temperature sensor uses the PTAT generator topology to calculate the temperature of the core. Temperature is defined by the following equation:

$$T = VPTAT \times M \times (q/k)$$

where:
VPTAT = voltage measured by the ADC.
M = gain determined by the analog circuit topology; in this case, a typical value of 462 µV/°C.
q = electron charge.
k = Boltzmann's constant.

TEMPERATURE SENSOR USAGE

The temperature sensor works with other blocks on the ADuCM350, and the following bits must be enabled and powered up fully before a temperature conversion can be made via the AFE configuration register (AFE_CFG):

- Bit 7: ADC_EN = 1.
- Bit 12: TEMP_SENSOR_EN = 1.

The temperature sensor is also selected via the ADC configuration register (AFE_ADC_CFG):

- Bits[4:0]: MUX_SEL = 00011 to select the temperature sensor ADC multiplexer input to the ADC.

The temperature conversion is initiated by setting Bit 13 of the AFE configuration register (AFE_CFG).

- Bit 13: TEMP_CONV_EN = 1.

After measurements are taken, an interrupt is flagged in the analog capture interrupt register:

- Bit 3: TEMP_RESULT_READY.

The results of the measurement can be accessed from the temperature sensor result register in 16-bit format.

The temperature measurement can use the 160 kSPS ADC output, sinc2hf output, or sinc2lf output, depending on the status of SUPPLY_LPF_EN in the AFE_CFG register and BYPASS_SUPPLY_LPF in the AFE_SUPPLY_LPF_CFG. The recommended setting uses the sinc2lf output.

The total measurement time from powering on the temperature sensor to getting a result, using the supply rejection filter, is covered in the AFE Example Use Cases section.

The temperature conversion (TEMP_CONV_EN = 1) converts the ADC conversion result into a temperature result, using the gain correction factor measured during Analog Devices factory calibration. This can be bypassed and calculations can be performed in software, in which case TEMP_CONV_EN can be 0.

The following formula converts the ADC result into the temperature result:

$$Temperature = ((Code - 2^{15}) \times Gain)/2^{14}$$

where:
Code is the ADC result (for example, read from AFE_SUPPLY_LPF_RESULT).
Gain is the temperature sensor correction factor read from TEMP_SENS_GAIN in the AFE_TEMP_SENS_TRIM register.

TEMPERATURE SENSOR MEASUREMENT DATA FORMAT

The temperature sensor result data from the ADC is stored in 16-bit twos complement format, with the MSB being the sign bit and with one LSB of the temperature result corresponding to 0.08°C.

Table 554. Temperature Sensor Measurement Data Format

Code	Temperature
0000 0000 0000 0000 (0 decimal)	−273.15°C
0000 0111 0001 0110 (1814 decimal)	−128°C
0000 1100 0101 1100 (3164 decimal)	−20°C
0000 1101 0101 0110 (3414 decimal)	0°C
0000 1110 1000 1111 (3727 decimal)	+25°C
0001 0000 1100 1110 (4302 decimal)	+70°C
0001 0011 1001 0110 (5014 decimal)	+128°C

TEMPERATURE SENSOR GAIN AND OFFSET CALIBRATION

The result of the temperature measurement is stored in the temperature sensor result register (AFE_TEMP_SENSOR_RESULT). Because both positive and negative temperatures can be measured, the temperature data is stored in twos complement format, as shown in Table 554.

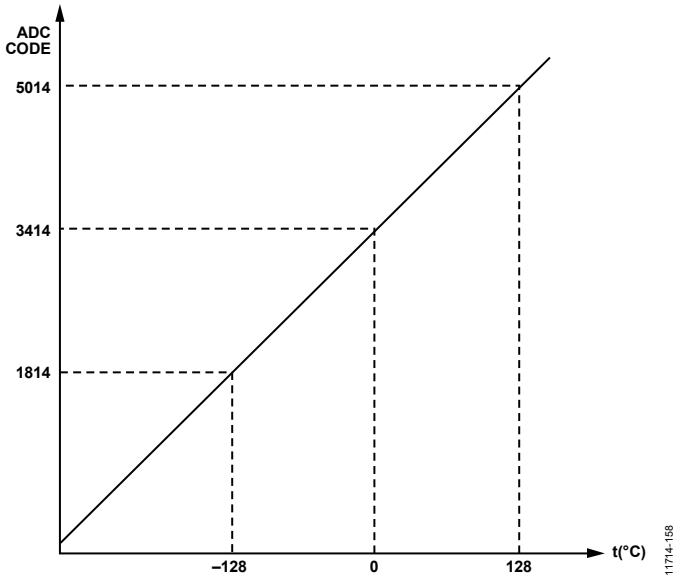


Figure 158. ADC Code vs. Temperature Range

Linear system thermal calibration is enabled to give users the flexibility to modify the gain and offset of the temperature sensor to take the specific thermal characteristics of their application design into account as follows:

$$AFE\_TEMP\_SENSOR\_RESULT = AFE\_ADC\_RESULT \times AFE\_ADC\_GAIN\_TEMP\_SENS + AFE\_ADC\_OFFSET\_TEMP\_SENS$$

where:
AFE_ADC_GAIN_TEMP_SENS is a 16-bit word that enables modification of the temperature sensor gain.
AFE_ADC_OFFSET_TEMP_SENS is a 16-bit, twos complement modification of temperature sensor offset.
AFE_TEMP_SENSOR_RESULT is a 16-bit, twos complement result of temperature sensor conversion.

Users can characterize their particular application during product development and set these values in the registers before performing temperature measurements because the values in the AFE_ADC_GAIN_TEMP_SENS and AFE_ADC_OFFSET_TEMP_SENS registers are used in calculating and storing the result in the AFE_TEMP_SENSOR_RESULT register. This function can also be useful for system development purposes.

Details about the proposed calibration routine for the internal temperature sensor on the AFE power-up are provided in the No Factory Calibration section.

CAPACITIVE TOUCH INTERFACE

INTRODUCTION

The ADuCM350 CapTouch subsystem interfaces with up to six capacitive touch buttons measured in self-capacitance mode and incorporates high performance capacitance sensing circuitry without external components.

The sensor input configuration is very flexible and uses several techniques to ensure that there are no false touches (that is, no registering touches due to a changing environment) on the external sensors. To minimize noise pickup from the system, the ADuCM350 CapTouch core includes several algorithms, such as median and averaging filtering measurements, as well as configurable excitation frequency and duty cycle.

The subsystem includes a self timer and touch and release routines to optimize the power consumption and reduction of the computing workload in the Cortex-M3.

The communication with the ADuCM350 ARM Cortex-M3 is provided by means of an AHB interface and interrupt, as shown in Figure 159.

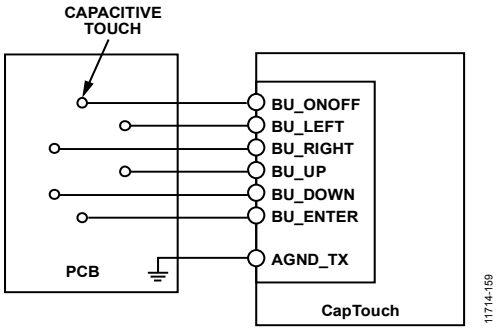


Figure 159. External Interface

INTERNAL BLOCK DIAGRAM

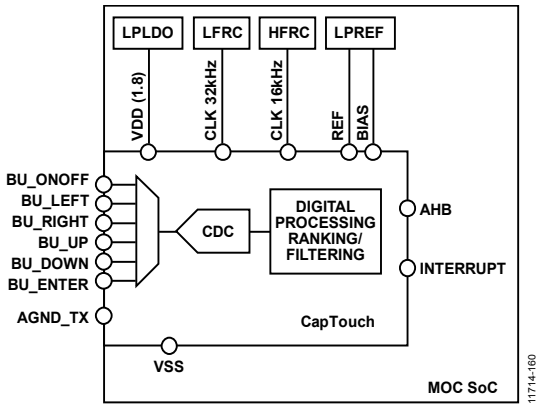


Figure 160. CapTouch Top Level Block Diagram

Table 555. Pin Description

Pin Name	Internal/External	Description
BU_ON_OFF	External	Capacitance Sensor Input 0
BU_LEFT	External	Capacitance Sensor Input 1
BU_RIGHT	External	Capacitance Sensor Input 2
BU_UP	External	Capacitance Sensor Input 3
BU_DOWN	External	Capacitance Sensor Input 4
BU_ENTER	External	Capacitance Sensor Input 5
AGND_TX	External	CapTouch PCB ground
VDD	Internal	Digital/analog power, 1.6 V to 2 V
VSS	Internal	Digital/analog ground
CLK32kHz	Internal	32 kHz clock
CLK16MHz	Internal	16 MHz clock
AHB[31:0]	Internal	AHB interface
Interrupt	Internal	Interrupt output

Note that the capacitance to voltage (C2V) sample algorithm, ADC sample rate, filtering, update rates, and thresholds are all controlled by registers preprogrammed through the AHB interface.

THEORY OF OPERATION

The CapTouch measures a change in the external sensor capacitance, converts the capacitance input signal into a digital value, and preprocesses/filters the data before writing to the results registers via the AHB interface, as shown in Figure 161.

The CapTouch provides up to five different interrupt events.

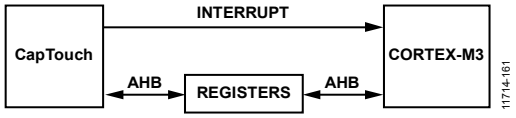


Figure 161. Digital Interface Connection

The sensing method measures the capacitance between an electrode and ground, usually the user's finger, and the measured output is processed by the digital filters to minimize spurious noise. The filtered values are stored in the result registers and subsequently can be processed by the ARM Cortex-M3 core.

Additionally, the CapTouch provides touch-and-release algorithms to autonomously monitor the inputs, improving the current consumption by reducing the workload in the Cortex-M3 core.

To perform the measurements, the CapTouch first outputs an excitation signal to charge the plate of the capacitor. When the user comes close to the sensor, a virtual capacitor is formed, with the user's finger acting as the second plate, as shown in Figure 162. In this case, there is an increase in capacitance that is measured by the capacitance to the digital converter.

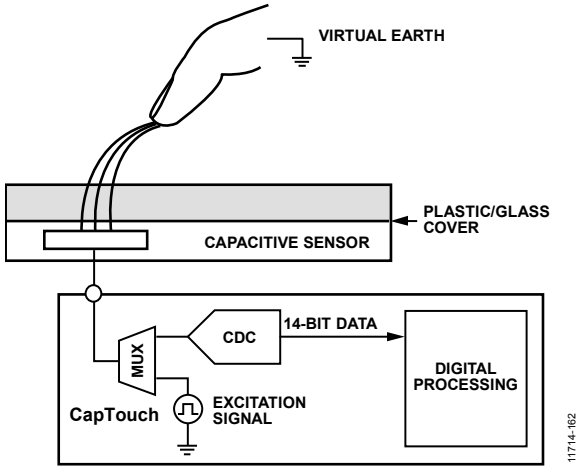


Figure 162. Capacitance Measurements Simplify Diagram

Capacitance to Digital Converter

The capacitance to digital converter (CDC) block converts a sensor capacitance into digital codes that are processed by the digital block. The CapTouch drives and measures on each sensor electrode. To enable the pins as CapTouch inputs, the DIG_CTOV_CAPTOUCH_EN bits (CT_CFG3[10:5]) must be set. However, prior to enabling these bits, ensure that the GPCON register is correctly configured to avoid internal short circuits that may damage the block. This block includes the following, as shown in Figure 163:

- LP filter: a low-pass filter removes high frequency noise coupled onto the input of the C2V circuit.
- C2V circuit: this circuit is formed by an excitation signal and an integrator that converts the capacitance charge from the BU_x inputs into the voltage.
- MUX: there are six possible inputs to the C2V, BU_x, that are connected to the input of the converter.
- PGA: this is the programmable gain amplifier for signal conditioning.
- CapDAC: this cancels out any parasitic or stray capacitances.
- Analog-to-digital converter (ADC): the ADC has 14 bits and is tested for −60 dB SNR.

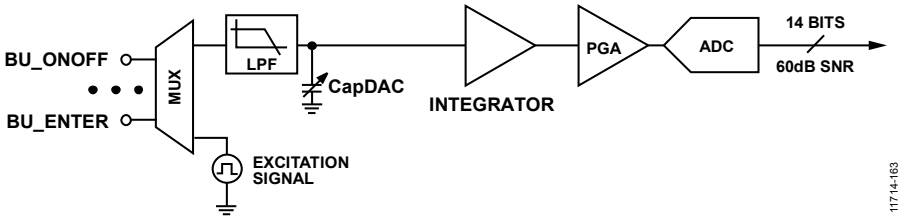


Figure 163. Capacitance to Digital Converter (CDC)

Low-Pass Filter

An extra input capacitance can be enabled in the input pin. This extra capacitance improves the low-pass filter transfer function response on the input of the C2V converter, removing high frequency noise on the BU_x inputs.

The capacitance is off by default. To turn on the capacitance, set the C2V_LPF bit (the CT_CFG2 register, Bit 31) to 1.

C2V Circuit

The C2V circuit is based on the charge transfer concept. This device employs a 4-phase C2V conversion scheme to measure the capacitance, as shown in Figure 164.

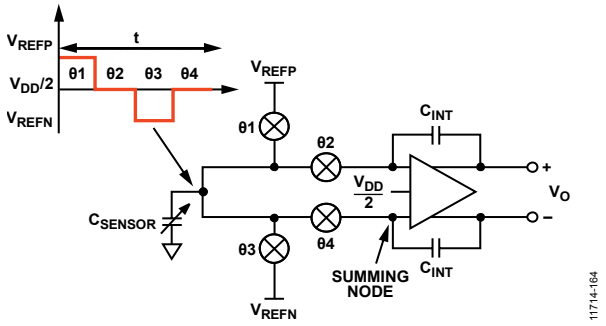


Figure 164. Capacitance to Voltage Converter

On Phase θ1, the unknown sensor capacitor (C_{SENSOR}) is connected to the voltage source (V_{REFP}), accumulating charge on that sensor capacitor. On Phase θ2, C_{SENSOR} is connected to the upper integrator summing node, and then the charge stored in C_{SENSOR} is discharged into the integrator capacitor (C_{INT}). The V_{REFP} output for capacitance sensors is as follows:

$$V_{OP} = \frac{C_{SENSOR}}{C_{INT}} \times \left(V_{REFP} - V_{DD}/2\right)$$

The integrator input terminals are held at $V_{DD}/2$. To cancel out $V_{DD}/2$, two additional phases—Phase θ3 and Phase θ4—are included in the circuit. On Phase θ3, C_{SENSOR} is connected to a second voltage source (V_{REFN}), and the charge is accumulated on C_{SENSOR}. On Phase θ4, C_{SENSOR} is connected to the opposite summing node so that the charge in C_{SENSOR} is discharged into C_{INT}. The following formula shows the total charge transferred into the integrator for the capacitance sensors after Phase θ1, Phase θ2, Phase θ3, and Phase θ4:

$$V_O = \frac{C_{SENSOR}}{C_{INT}} \times \left(V_{REFP} - V_{REFN}\right)$$

Additionally, to minimize parasitic loads and noise, it is possible to excite the other capacitive inputs at the same time that a BU_x is measured. The CapTouch offers four options programmed by the AIN_SEL bits (CT_CFG1[23:22]). These options are summarized in Table 556.

Table 556. Excitation Options for Pads

Code	Option
00	Inputs connected to the same excitation signal (preferred option)
01	Inputs connected to $V_{DD}/2$
10	Inputs connected to GND
11	Inputs floating

C2V Configuration

Capacitance Measurements Timing

Figure 165 shows the timing diagram for the 4-phase C2V capacitance conversion, which allows the user to configure the excitation frequency and the duty cycle.

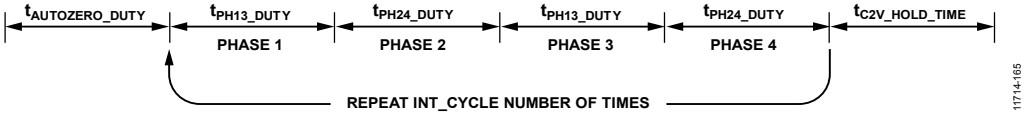


Figure 165. Capacitance Measurement Timing

The phase times are register programmable via the CFG2 register. The phase times for Phase θ1 and Phase θ3 are programmable from 0.25 µs to 8 µs via the PH13_DUTY bits (CT_CFG2[4:0]). The phase times for Phase θ2 and Phase θ4 are programmable from 62.5 ns to 16 µs via the PH24_DUTY bits (CT_CFG2[12:5]).

The automatic zero time, which is the time taken by the C2V converter to discharge the capacitive sensor before each measurement, is also programmable. The automatic zero time can be set to values from 2 µs to 9 µs via the AUTOZERO_DUTY bits (CT_CFG1[12:10]).

The number of integration cycles taken by the C2V converter is programmable via the INT_CYCLE bits (CT_CFG1[28:26]). For capacitance measurements, the C2V converter can integrate within one cycle to 16 cycles. The time taken for one integration cycle is

$$(2 \times PH13\_DUTY) + (2 \times PH24\_DUTY)$$

The C2V converter hold time, which occurs after the C2V converter has finished all integration cycles, is programmable from 0.25 µs to 8 µs via the C2V_HOLD_TIME bits (CT_CFG2[26:22]).

Integrator Capacitor Range, ΔC_{INT}

The integrator capacitor range of the C2V converter is programmable for each stage, and the C2V_IP_RANGE bits (CT_STAGEx_CFG[19:16]) control the ΔC_{INT} range. The input range can be programmed from ±0.58275 pF to ±9.324 pF.

Bias Current

The bias current for the C2V converter is programmable from 6 µA to 13 µA. The C2V_BIAS bits (CT_CFG1[2:0]) control the bias current. The default bias current is 10 µA.

Capacitance Compensation, CapDAC

The typical background capacitance to ground can be approximately 20 pF to 30 pF. Changes in capacitance to ground due to the proximity of the user's finger are typically hundreds of femtofarads. This small capacitance variance drives the requirement to compensate for the background capacitance, maximizing the output dynamic range and allowing higher sensitivity (measured in V/pF) in the CDC block.

This background parasitic load must be discarded to obtain accurate output values. By using a CapDAC scheme, the CapTouch compensates such parasitic capacitance. The CapDAC can be understood as a negative capacitance that is internally connected to C_{SENSOR}. The CapDACs are applied with unique settings for each stage.

A programmable capacitor (CDAC) is connected in parallel with the sensor, as shown in Figure 166. The CDAC capacitor is driven by an excitation signal that is in antiphase with the excitation to the sensor. The effect is that a charge proportional to CDAC is subtracted from the measured C_{SENSOR} capacitance.

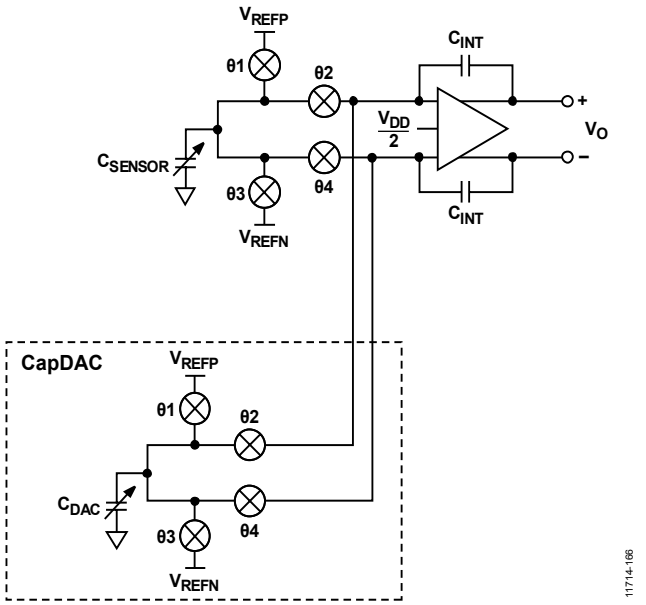


Figure 166. Capacitance Compensation, CapDAC

Programmable Gain Amplifier

The ADuCM350 CapTouch includes a programmable gain amplifier to increase the sensitivity in case it is needed. The gain is completely register programmable for each stage. The gain amplifier increases signal amplitude into the converter by 2, 4, 8, or 16. By setting PGA_GAIN (CT_STAGEx_CFG[21:20]), users can set the stage gain.

The bias current for the gain amplifier is programmable from 3 µA to 6.5 µA. The PGA_BIAS bits (CT_CFG1[5:3]) control the bias current. The default bias current is 5 µA.

The gain amplifier can be bypassed and powered down. To bypass the gain amplifier for all capacitance measurements, set the BYPASS_GAIN bit (CT_CFG1[8]) to 1. The gain amplifier can also be bypassed on a per stage basis. To bypass the gain amplifier for particular measurements, set PGA_BYPASS (CT_STAGEx_CFG[8]) to 1 in the relevant register. The gain amplifier is automatically powered down when not in use to reduce power consumption.

It is recommended to set the BYPASS_GAIN bit unless the amplifier is strictly necessary.

ADC

The analog-to-digital conversion is performed by a 14-bit successive approximation register (SAR) ADC. The 14-bit SAR ADC converts the voltage values to digital. The SAR provides −60 dB SNR. The converted data is processed in the digital block that provides rank and averaging filters, which can be used to filter out any spurious noise.

Digital Block

In a CapTouch application, noise can couple into the analog front-end measurement. Different stages can be enabled to provide a higher noise reduction, as shown in Figure 167.

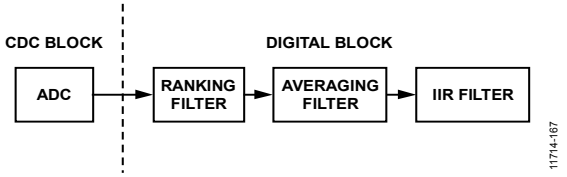


Figure 167. Digital Block

To eliminate impulse noise, rank, average, and infinite impulse response (IIR) filtering are implemented in the CapTouch block. The rank filter ranks all the samples from the CDC within a data array, from smallest to largest value. Out of range samples end up either at the top or at the bottom of the array. The CapTouch implements an averaging function where a selectable number of samples from the ranked data array are averaged. The averaging filter allows averaging samples from the middle of the data array; therefore, the out of range samples (impulse noise) are not included in the final averaged number, as shown in Figure 168.

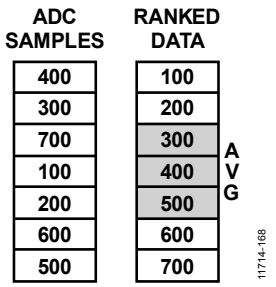


Figure 168. Rank and Average Filtering Example

Finally, the IIR filter processes the data to minimize any spurious noise, and the results are stored in the CT_CDC_RESx registers after being processed.

The CT_CDC_RESx registers can be configured to store the difference between the CDC value and the baseline value, instead of storing the CDC value. This is used in certain positional calculation algorithms. The RES_SEL bit (Bit 13 of the CT_CFG3 register) controls the data stored in the CT_CDC_RESx registers.

Rank Filter

The rank filter suppresses the isolated out of range noise and sets the number of measurements taken. Each measurement is arranged in a temporary array, where the first value is the smallest measurement, and the last value is the largest measurement. The number of samples to be ranked is set by the RANK_FILT bits in the CT_AVG register; 4, 8, 16, or 32 samples can be taken for each measurement.

Setting the RANK_FILT bits to 0 disables the rank filter, and the number of samples is selected by the averaging filter bits (AVG_GROUP1, CT_AVG[18:16]).

Averaging Filter

The averaging filter takes the ranked data, if the rank filter has been enabled, from the data array and performs averaging on the selected data elements. Two groups of data elements can be averaged separately, as shown in Figure 169. This enables filtering of different types of noise. One or two groups can be selected for averaging. If only one group is selected for averaging, the Group 2 settings are ignored.

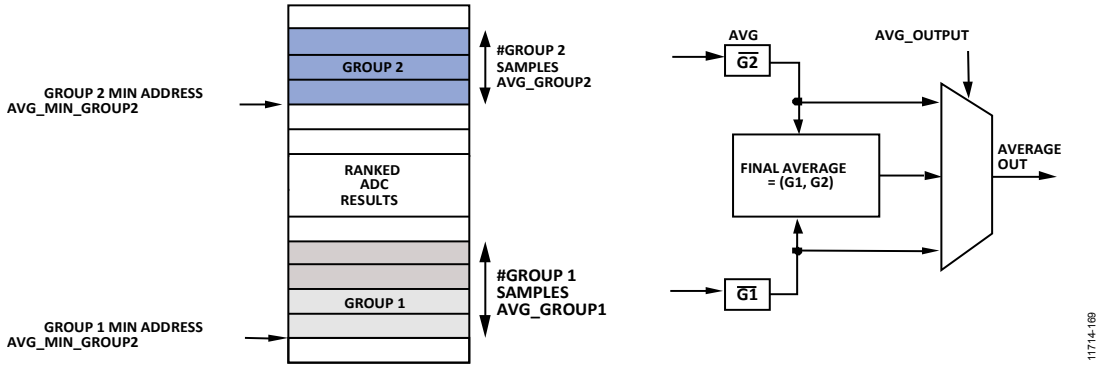


Figure 169. Averaging Filter

For each sample averaging group, the address where the averaging starts and the number of samples to average can be configured. The CT_AVG register controls the averaging and the group selected.

AVG_MIN_GROUP1 sets the minimum rank position for data to be averaged in Group 1, and AVG_GROUP1 sets the number of samples to be averaged. AVG_MIN_GROUP2 sets the minimum rank position for data to be averaged in Group 2, and AVG_GROUP2 sets the number of samples to be averaged.

The AVG_OUTPUT bits select the final averaged output value to be stored in the CT_CDC_RESx registers. These bits can be used to select no averaging, average of Group 1, average of Group 2, or average of Group 1 and Group 2 for this value.

IIR Filter

The CDC IIR filter implements a low-pass filter response. This filter requires little data memory and does not need multiple conversion results to calculate the filtered output. The current result is compared with the previous result.

The IIR filter can be enabled globally for measurements by setting IIR_EN in the CT_CFG3 register to 1. The filter coefficient weight for measurements can be chosen via the IIR_WEIGHT bits. The filter weight can be set from 0 to 8, CT_CFG3[4:1]. The filter transfer function is defined as follows:

$$CURRENT\_RESULT = ((PREVIOUS\_RESULT \times NEW\_SAMPLE\_WEIGHT) + (CURRENT\_SAMPLE) \times (16 - NEW\_SAMPLE\_WEIGHT)))/16$$

CapTouch Sequencer

The CapTouch has an integrated sequencer to implement conversion control for the input channels, BU_x. The sequencer represents a succession of conversions performed when the CapTouch receives a start of conversion signal until the data is processed.

For each conversion stage, a different BU_x can be converted by configuring the CT_STAGEx_CFG registers, where x indicates the current stage number, from 0 to 15. The CIN_CON_POS_CDC bits, CT_STAGEx_CFG[26:24], control which BU_x is connected to the CapTouch connection for a stage, as shown in Figure 170.

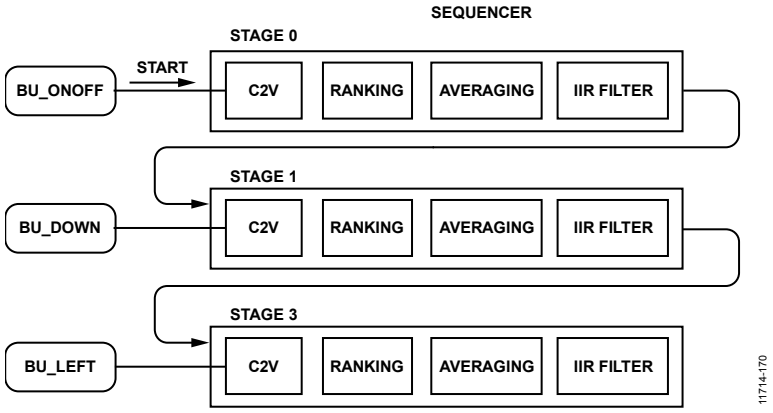


Figure 170. CapTouch Sequencer Example

Each stage can be individually optimized, using the register settings. Each conversion stage is uniquely configured to support multiple capacitance sensor interface requirements. Per stage configurability is a key feature in providing a robust sensor response for different types of buttons.

A conversion sequence is defined as a sequence of CapTouch conversions starting at Stage 0 and ending at Stage 15. The CIN_CON_POS_CDC_EN bit (CT_STAGEx_CFG[27]) must be set to 1 to enable the connection configured in the CIN_CON_POS_CDC bits. If the enable bit is not set to 1, the particular stage is ignored.

While the CapTouch is converting/processing data, the SW_STATUS bit (Bit 3 of CT_CDC_PWR) is set to 1.

The number of required conversion stages depends solely on the number of sensors attached. Additional stages can be set up for noise measurement on individual BU_x sensors.

How the CapTouch Sequencer Works

For each Conversion Stage x in the sequence, the CT_STAGEx_CFG register must be configured and the CT_MEAS_SEL register specific bit must be enabled. Each Bit x in this register indicates whether Stage x is enabled in the CapTouch sequencer. The GND_SEL bits (CT_MEAS_SEL[15:0]) allow the individual selection of each stage in the CapTouch sequencer.

Specific configuration settings are downloaded from the registers for each stage. If the rank or averaging filters are enabled, each conversion is repeated a certain number of times, depending on the values programmed for the filters, as shown in Figure 171. Setting the CONVERSION_RESET bit (CT_BASELINE_CTRL[28]) to 1 resets the conversion sequence to Stage 0.

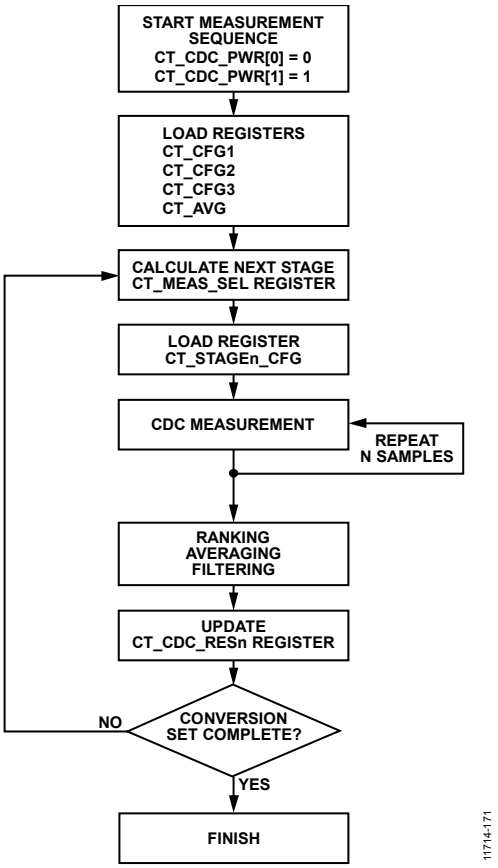


Figure 171. CapTouch Sequencer

Preprogrammed Functions

Additionally, the digital block contains preprogrammed functions to reduce the computational load in the Cortex-M3. The CT_CAL_EN register globally enables these functions.

Baseline Value Tracking

Environmental effects, such as temperature and humidity, cause the baseline value to drift over time. These changes must be tracked to provide a reliable reference for sensor touch response and activation thresholds, which are calculated based on this baseline (see the Baseline Calibration section for more details).

Proximity Detection (Fast and Slow Detection)

To prevent the baseline reference level from tracking the CDC response as the user approaches the sensor, the CapTouch uses a proximity detection algorithm to detect this occurrence and halt updates to the baseline. The algorithm must be able to distinguish between fast and slow (hovering) approaches from the user, and it must be immune to changes in the environment that may cause changes in the response. The proximity block contains two IIR filters, fast and slow, to allow for filtering of the processed value. These two filters are IIR implementations and allow for monitoring of the baseline and detection of proximity, as shown in Figure 172.

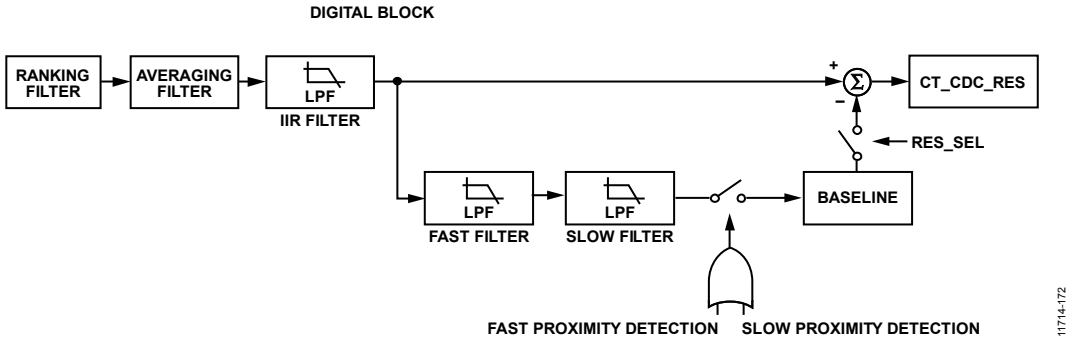


Figure 172. Proximity Detection Block

Essentially, fast proximity reacts quickly to changes in the CDC value (such as when a user moves toward the sensor). The sensitivity of the fast filter is set by the sampling frequency bits (FAST_FILTER_UPDATE) and the sample coefficient weight bits (CDC_COEFF). All of these bits can be found in the CT_BASELINE_CRTL register. The filter transfer function is defined as follows:

CURRENT_RESULT = ((*PREVIOUS_RESULT* × *NEW_SAMPLE_WEIGHT*) + (*CURRENT_SAMPLE* × (16 – *NEW_SAMPLE_WEIGHT*))/16

The fast proximity detection output rises when the difference between the new sample from the IIR filter and the previous fast filter result is higher than the threshold level set by the FAST_PROX bits, (CT_BASELINE_CRTL[23:16]), as described in the following equation:

4 × *FAST_PROX* < *new_sample* – *previous_result*

The sensitivity of the slow filter is set by the filter coefficient weight via the CDC_COEFF bits. The filter weight can be set from 0 to 16 using BL_COEFF bits (CT_BASELINE_CRTL[3:0]). The filter transfer function for the slow filter is defined as follows:

CURRENT_RESULT = ((*PREVIOUS_RESULT* × *NEW_SAMPLE_WEIGHT*) + (*CURRENT_SAMPLE* × (16 – *NEW_SAMPLE_WEIGHT*))/16

The slow proximity detection rises when the difference between the new sample from the fast filter and the baseline value is higher than the threshold value programmed in the SLOW_PROX bits (CT_BASELINE_CRTL[15:8]), as described in the following equation:

4 × *SLOW_PROX* < *NEW_SAMPLE* – *Baseline*

If the fast proximity detection has generated an event, the baseline value updated is delayed via a programmed register, that is via the BASELINE_CAL_DELAY bits, CT_BASELINE_CRTL[31:29].

An example of the proximity detection algorithm's operation is shown in Figure 173.

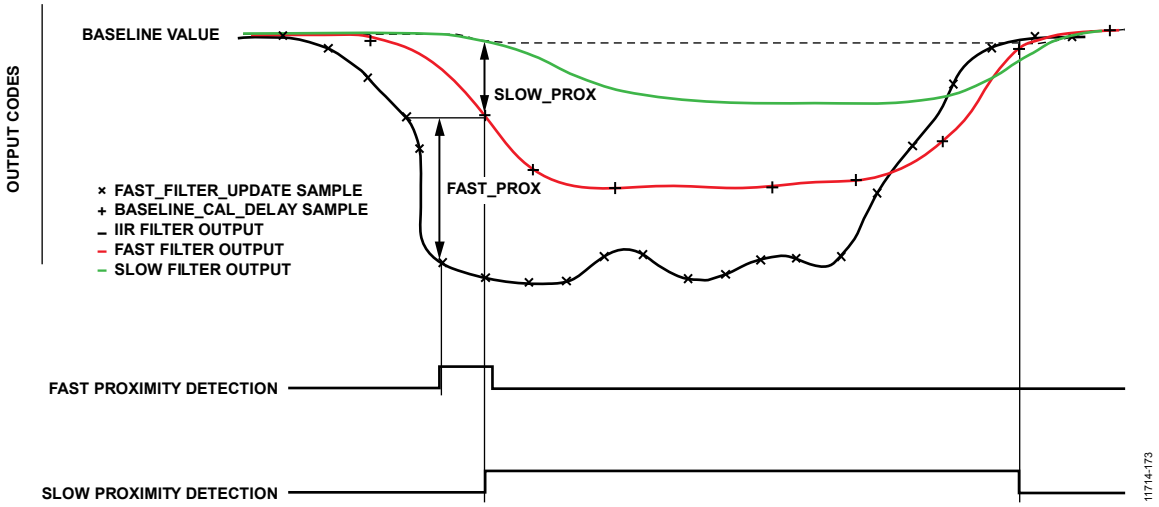


Figure 173. Proximity Detection Event

CapTouch Self Timer

The CapTouch self timer allows the capacitive sensor to be scanned automatically at programmable intervals. The programmable delay is configurable via the CT_CFG2 register. The self timer runs from the system 32 kHz clock. This functionality reduces the power consumption as the Cortex-M3 core can be kept in SYS_SLEEP and awake via an internal interrupt.

To enable the self timer, the STIMER_EN bit, CT_CFG[27], must be set. The time is configured by the SELF_TIMER_WAIT (CT_CFG2[21:13]) value in 1.953 ms steps up to 998.05 ms.

Touch and Release Detection

The CapTouch provides touch and release detection algorithms that minimize the Cortex-M3 workload, processing the measurements directly in the CapTouch subsystem. These algorithms must be used in conjunction with the self timer for optimum operation.

The touch algorithm can be enabled individually by setting the TCH_DETECT_ENABLE bit (CT_TOUCH_CFG2[15]). The release algorithm is enabled by setting the REL_DETECT_ENABLE bit (CT_RELEASE_CFG2[15]) and generates a valid event only if a previous touch event has been detected. This implies that the touch algorithm must be enabled as well.

The algorithm can be configured to operate with different inputs, as described in Table 557. The touch algorithm counts the number of times that the difference between the selected inputs has been lower (or higher) than a preprogrammed threshold set by the TCH_UPPER_THLD bits (CT_TOUCH_CFG1[29:16]). To improve the noise immunity, a second threshold level is provided, TCH_LOWER_THLD. If the output code is in between the two threshold levels, the count is not incremented and does not reset. If the code crosses the lower hysteresis level determined by TCH_LOWER_THLD, the count is reset.

Table 557. Touch and Release Algorithm Input Options

Bits	Input	Description
000	Load all 0s	Load 0s
001	Load CDC value	Averaging filter output
010	Load slow filter output	CDC IIR filter
011	Load fast filter output	Fast filter output
100	Load ambient value	Baseline value

An example of operation is shown in Figure 174. In this case, the algorithm has been configured to compare the CDC IIR filter output with the baseline value. Note that an increment in the output capacitance generates a decrement in the number of codes.

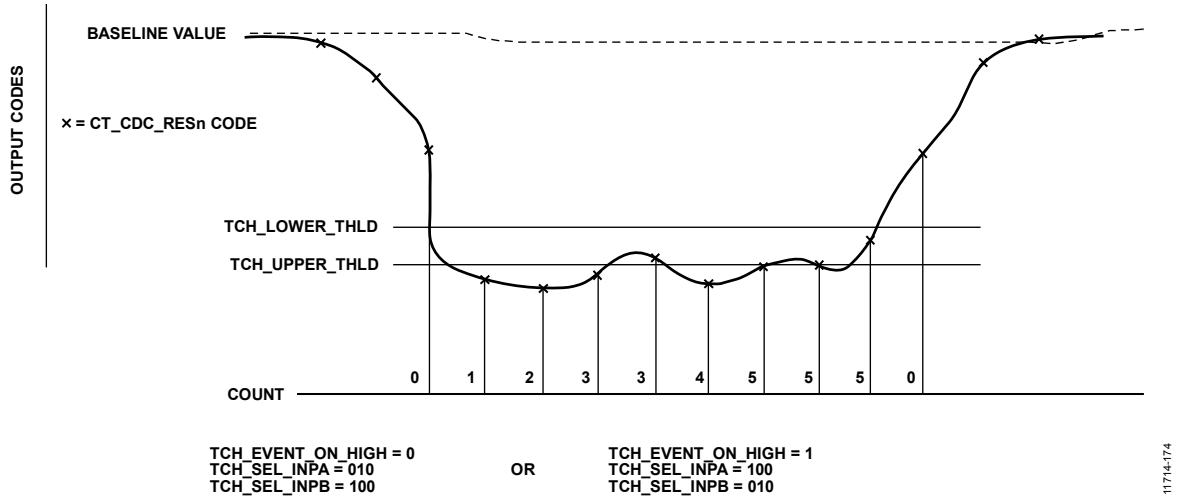


Figure 174. Touch Detection Example

The release algorithm works in similar way.

Error Detection

To detect calibration/measurements errors, the CapTouch provides threshold registers, CT_SENSOR_THR_CFGx, which must be configured by the user. These registers enable an individual threshold for any of the CapTouch sequence.

The error detection algorithm compares the absolute difference between the CDC IIR filter and the baseline value and generates an error condition if the value is higher than the threshold level.

There are two cases than can generate the error:

1. Baseline calibration error: the baseline value is calibrated when the user presses the button.
2. Saturation error: the gain in any of the stages is too high and the ADC can become saturated.

Note that the threshold voltage is common for both errors, but one error is positive and the other error is negative.

The OFF_HSTAT bits (CT_OFFS_HSTAT[15:0]) provide the error status of the baseline calibration error.

OFF_LSTAT bits (CT_OFFS_LSTAT[15:0]) provide the error status of the saturation error.

Interrupts

The CapTouch block provides one hardware interrupt connection to the core. The interruption can be generated by up to five different sources, which can be enabled individually by the CT_IEN register. The interrupt is always generated at the end of a conversion sequence. The Cortex-M3 polls different registers to find the exact sequence that has generated the interruption, as summarized in Table 558.

Table 558. Interruption Sources

Interruption Source	CT_IEN Bit	CT_INT Poll	Sensor Source	Comments
Release Detection	RELEASE_DETECTED_IEN	RELEASE_DETECTED	CT_TOUCH_STAT[31:16]	Release detection
Touch Detection	TOUCH_DETECTED_IEN	TOUCH_DETECTED	CT_TOUCH_STAT[15:0]	Touch detection
Proximity Detection	PROX_DETECTED_IEN	PROX_DETECTED	CT_PROX_STAT[15:0]	Halt baseline update (slow or fast)
			CT_FPROX_STAT[15:0]	Fast proximity detection
Error Detection	STATUS_GT_THRESHOLD_IEN	STATUS_GT_THRESHOLD	CT_OFFS_LSTAT[15:0]	Baseline calibration error
			CT_OFFS_HSTAT[15:0]	Saturation error
End Conversion	CONV_SET_COMPLETE_IEN	CONV_SET_COMPLETE		End sequence

The interrupt source must be cleared manually by setting the respective bit of the interruption source in CT_INT.

If an interruption has been generated and the bit is not cleared manually, the interruption generator ignores any future interruptions from the same source.

Baseline Calibration

In capacitance sensing applications, a baseline is required to set a reference level for all capacitance measurements. The reduction of the baseline offset is done using the CapDACs.

A simplified block diagram in Figure 175 shows how to apply the offset registers to null the baseline offsets.

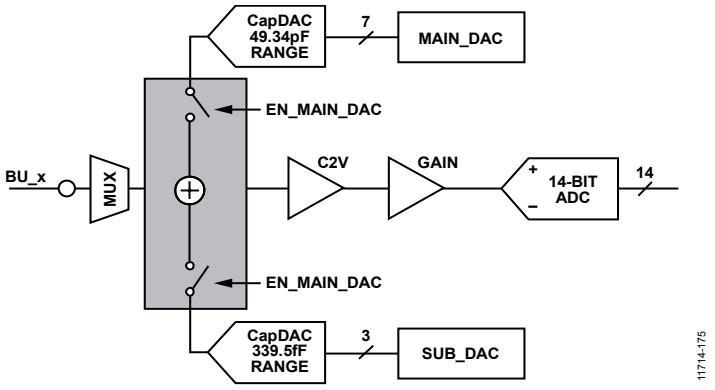


Figure 175. Applying the CapDACs

The 7-bit MAIN_DACn and 3-bit SUB_DACn (in Register CT_STAGE0_CFG to Register CT_STAGE15_CFG) program the offset DACs to provide 48 fF resolution offset adjustment over a range of ~50 pF. The offset value can be applied to the CDC or can be disabled. The EN_MAIN_DAC bits have effect if the MAIN_DACn bits and/or SUB_DACn bits are enabled.

All offset DACs are independently programmable for each stage in the CT_STAGEx_CFG registers.

The best practice is to ensure that the CDC output for any stage is approximately equal to ¾ scale (~12,000 codes) when all sensors are inactive. To correctly offset the parasitic capacitance for each stage, use the following procedure:

1. Read back the CDC value from the CT_CDC_RESx register.
2. If the value is not close to midscale, enable the main DAC and increase the value of MAIN_DACn.
3. If the CDC value in CT_CDC_RESx is closer to midscale, repeat Step 2. If the CDC value is further from midscale, decrease the MAIN_DAC value by 1.
4. If MAIN_DACn has reached its maximum value and the CDC value is still not close to midscale, increase the value of SUB_DACn.
5. If the CDC value in CT_CDC_RESx is closer to midscale, repeat Step 4. If the CDC value is further from midscale, decrease the SUB_DACn value by 1.

The goal is to ensure that CT_CDC_RESx is as close to ¾ scale as possible, operating in the linear region of the ADC. This process is required only once during the initial capacitance sensor characterization or once after an overrange fault has been detected.

Baseline Recalibration

In the event of an error condition where a proximity or status flag is asserted due to a sensor or user fault, it may be necessary for system software to recalibrate the baseline algorithms based on the current CDC results. The FORCE_CAL bit (BASELINE_CTRL[27]) is used to reset the logic and IIR filter results. After recalibration is complete, this bit is cleared automatically.

Output Data Rate

The output data rate is the rate at which the CapTouch provides converted capacitance values from the CDC so that the touch states can be updated. In other words, the output data rate is the time required to provide valid CDC measurements.

The output data rate is determined by the total number of conversions performed by the CDC, the timing settings, and the sample to sample delay. The rank filter determines the number of samples per channel, and the capacitance measurements affect the total number of conversions required.

Usually a trade-off among output data rate, power, and noise is necessary, and the programmable features available on ADuCM350 CapTouch facilitate this trade-off.

The output data rate can be calculated as follows (terms are defined in Table 559 and Table 560):

$$OUTPUT\_DATA\_RATE = 1/t_{MEASUREMENTS}$$

$$T_{SAMPLE} = (t_{AZ} = t_{CLK} + (2 \times (t_{\Phi 13} + t_{\Phi 24} + t_{CLK}) \times INT\_CYCLE) + (2 \times t_{HOLD}) + (2 \times t_{CONV}))$$

$$t_{MEASUREMENT} = t_{CONV\_SET\_DELAY} + (((t_{SAMPLE} \times AVG\_SAMPLES) + t_{START} + t_{STOP}) \times (MEAS/1))$$

Table 559. Output Date Rate Calculation—Variables Glossary

Parameter	Register Bits	Description	Value
$t_{MEASUREMENTS}$		Number of capacitance measurements.	Up to 6
RANK_SAMPLES	RANK_FILT (CT_AVG[30:28])	Number of samples from rank filter. If this value is set to 1 (no ranking), the AVG_GROUP1 (AVG[18:16]) value must be substituted into the equation.	1, 4, 8, 16, or 32
$t_{CONV\_SET\_DELAY}$	SELF_TIMER_WAIT (CT_CFG2[21:13])	Delay time between the end of one conversion set and the start of the next. During this time, the CapTouch is powered down (see the CapTouch Self Timer section for more details).	0 ms to 1000 ms
t_{AZ}	AUTOZERO_DUTY(CT_CFG1[12:10])	CDC autozero time.	Min = 2 µs, max = 9 µs
$t_{\Phi 13}$	PH13_DUTY (CT_CFG2[4:0])	Stages time Phase 1 and Phase 3 duty.	Min = 0.25 µs, max = 8 µs
$t_{\Phi 24}$	PH24_DUTY (CT_CFG2[12:5])	Stages time Phase 2 and Phase 4 duty.	Min = 0.25 µs, max= 8 µs
t_{HOLD}	CV_HOLD_TIME (CT_CFG2[26:22])	CDC hold time.	Min = 0.25 µs, max = 8 µs
t_{CONV}		CDC conversion time.	1.2 µs
t_{START}		Time to work out next stage to be converted.	0.371 µs
t_{STOP}		Initial processing of Stage n.	0.783 µs
t_{CLK}		1 clock cycle = 1/16,000,000.	0.0625 µs
INT_CYCLE	INT_CYCLE (CT_CFG1[26:24])	Number of integration cycles.	1, 2, 4, 8, or 16

Table 560. Output Data Rate Example

Parameter	Value	Comments
$t_{MEASUREMENTS}$	6	Six buttons
AVG_SAMPLES	8	Eight samples per measurement
t_{AZ}	2 µs	
$t_{\Phi 13}$	0.5 µs	
$t_{\Phi 24}$	0.5 µs	
$t_{\Phi 12}$	0.5 µs	
$t_{CONV\_SET\_DELAY}$	0	Maximum speed, no delay between conversion sets
t_{HOLD}	1 µs	
t_{CONV}	1.2 µs	
INT_CYCLE	1	One integration cycle
$t_{MEASUREMENTS}$	360 µs	
Output Data Rate	2.77 kHz	Maximum repeat rate for sampling all six buttons

Table 561. Output Data Rate Example—Single Button Standby Mode

Parameter	Value	Comments
$t_{MEASUREMENTS}$	1	1 button
AVG_SAMPLES	8	Eight samples per measurement
t_{AZ}	2 µs	
$t_{\Phi 13}$	0.5 µs	
$t_{\Phi 24}$	0.5 µs	
$t_{\Phi 12}$	0.5 µs	
$t_{CONV\_SET\_DELAY}$	330ms	Maximum speed, no delay between conversion sets
t_{HOLD}	1 µs	
t_{CONV}	1.2 µs	
INT_CYCLE	1	One integration cycle
$t_{MEASUREMENTS}$	330 ms	
Output Data Rate	3 Hz	Repeat rate dominated by power down time (CONV_SET_DELAY)

Power Dissipation

The current consumed by the CapTouch varies according to the setup. Table 562 shows the current for some typical setup conditions. In each case, the C2V timing is the same and the PGA is not enabled.

Table 562. Power Dissipation Examples

Mode Description	Update Rate (Completed Sets Per Second)	Number of Buttons Monitored (Samples Per Set)	Amount of Averaging/Filtering (Conversions Per Sample)	Maximum CapTouch Current
Setup 1: Typical Standby Mode	3 Hz	1	8	350 nA
Setup 2: Six-Button Active Mode	50 Hz	6	8	25 µA
Setup 3: Six-Button Active Mode with Enhanced Filtering	50 Hz	6	16	50 µA
Setup 4: Six-Button Active Interface with 4× C2V Integration Cycles	50 Hz	6	8	30 µA
Setup 5: Six-Button Proximity Mode	3 Hz	6	8	1.5 µA

CAPACITIVE TOUCH MEMORY MAPPED REGISTERS

Capacitive Touch Register Map

Table 563. CapTouch Register Summary

Address	Name	Description	Reset	RW
0x40084000	CT_CDC_PWR	Power CDC control	0x00000000	RW
0x40084004	CT_CFG1	CapTouch Control Configuration Register 1	0x00000064	RW
0x40084008	CT_CFG2	CapTouch Control Configuration Register 2	0x00000000	RW
0x4008400C	CT_CFG3	AFE Control Configuration Register 3	0x00000000	RW
0x40084010	CT_MEAS_SEL	Capacitance Measurement Stage Selection	0x00000000	RW
0x40084014	CT_BASELINE_CTRL	Baseline Control Settings	0x00000000	RW
0x40084018	CT_AVG	Setup of the Rank-and-Average Filtering	0x00000000	RW
0x4008401C	CT_CAL_EN	Enable Calibration for Measurement Stages	0x00000000	RW
0x40084020	CT_TOUCH_CFG1	Touch Detection Thresholds	0x00000000	RW
0x40084024	CT_TOUCH_CFG2	Touch Detection Time-out	0x00000032	RW
0x40084028	CT_RELEASE_CFG1	Release Detection Thresholds	0x00000000	RW
0x4008402C	CT_RELEASE_CFG2	Release Detection Time-out	0x00000032	RW
0x40084030	CT_IEN	Interrupt Enable	0x00000000	RW
0x40084034	CT_INT	Primary Interrupt Register	0x00000000	RW1C
0x40084038	CT_OFFS_HSTAT	Stage Offset High Interrupt Status	0x00000000	R
0x4008403C	CT_OFFS_LSTAT	Stage Offset Low interrupt Status	0x00000000	R
0x40084040	CT_PROX_STAT	Stage Proximity Status	0x00000000	R
0x40084044	CT_FPROX_STAT	Stage Fast Proximity Status	0x00000000	R
0x40084048	CT_TOUCH_STAT	Stage Touch/Release Status	0x00000000	R
0x4008404C	CT_STAGE0_CFG	Stage 0 Configuration	0x00000000	RW
0x40084050	CT_STAGE1_CFG	Stage 1 Configuration	0x00000000	RW
0x40084054	CT_STAGE2_CFG	Stage 2 Configuration	0x00000000	RW
0x40084058	CT_STAGE3_CFG	Stage 3 Configuration	0x00000000	RW
0x4008405C	CT_STAGE4_CFG	Stage 4 Configuration	0x00000000	RW
0x40084060	CT_STAGE5_CFG	Stage 5 Configuration	0x00000000	RW
0x40084064	CT_STAGE6_CFG	Stage 6 Configuration	0x00000000	RW
0x40084068	CT_STAGE7_CFG	Stage 7 Configuration	0x00000000	RW
0x4008406C	CT_STAGE8_CFG	Stage 8 Configuration	0x00000000	RW
0x40084070	CT_STAGE9_CFG	Stage 9 Configuration	0x00000000	RW
0x40084074	CT_STAGE10_CFG	Stage 10 Configuration	0x00000000	RW
0x40084078	CT_STAGE11_CFG	Stage 11 Configuration	0x00000000	RW
0x4008407C	CT_STAGE12_CFG	Stage 12 Configuration	0x00000000	RW
0x40084080	CT_STAGE13_CFG	Stage 13 Configuration	0x00000000	RW
0x40084084	CT_STAGE14_CFG	Stage 14 Configuration	0x00000000	RW
0x40084088	CT_STAGE15_CFG	Stage 15 Configuration	0x00000000	RW
0x4008408C	CT_SENSOR_THR_CFG0	Stage 0 and Stage 1 Sensor Threshold	0x00000000	RW
0x40084090	CT_SENSOR_THR_CFG1	Stage 2 and Stage 3 Sensor Threshold	0x00000000	RW
0x40084094	CT_SENSOR_THR_CFG2	Stage 4 and Stage 5 Sensor Threshold	0x00000000	RW
0x40084098	CT_SENSOR_THR_CFG3	Stage 6 and Stage 7 Sensor Threshold	0x00000000	RW
0x4008409C	CT_SENSOR_THR_CFG4	Stage 8 and Stage 9 Sensor Threshold	0x00000000	RW
0x400840A0	CT_SENSOR_THR_CFG5	Stage 10 and Stage 11 Sensor Threshold	0x00000000	RW
0x400840A4	CT_SENSOR_THR_CFG6	Stage 12 and Stage 13 Sensor Threshold	0x00000000	RW
0x400840A8	CT_SENSOR_THR_CFG7	Stage 14 and Stage 15 Sensor Threshold	0x00000000	RW
0x400840AC	CT_CDC_RES0	Stage 0 and Stage 1 Results	0x00000000	R
0x400840B0	CT_CDC_RES1	Stage 2 and Stage 3 Results	0x00000000	R
0x400840B4	CT_CDC_RES2	Stage 4 and Stage 5 Results	0x00000000	R
0x400840B8	CT_CDC_RES3	Stage 6 and Stage 7 Results	0x00000000	R
0x400840BC	CT_CDC_RES4	Stage 8 and Stage 9 Results	0x00000000	R
0x400840C0	CT_CDC_RES5	Stage 10 and Stage 11 Results	0x00000000	R

Address	Name	Description	Reset	RW
0x400840C4	CT_CDC_RES6	Stage 12 and Stage 13 Results	0x00000000	R
0x400840C8	CT_CDC_RES7	Stage 14 and Stage 15 Results	0x00000000	R
0x400840CC	CT_BASELINE0	Stage 0 Fast Filter and Baseline Results	0x00000000	R
0x400840D0	CT_BASELINE1	Stage 1 Fast Filter and Baseline Results	0x00000000	R
0x400840D4	CT_BASELINE2	Stage 2 Fast Filter and Baseline Results	0x00000000	R
0x400840D8	CT_BASELINE3	Stage 3 Fast Filter and Baseline Results	0x00000000	R
0x400840DC	CT_BASELINE4	Stage 4 Fast Filter and Baseline Results	0x00000000	R
0x400840E0	CT_BASELINE5	Stage 5 Fast Filter and Baseline Results	0x00000000	R
0x400840E4	CT_BASELINE6	Stage 6 Fast Filter and Baseline Results	0x00000000	R
0x400840E8	CT_BASELINE7	Stage 7 Fast Filter and Baseline Results	0x00000000	R
0x400840EC	CT_BASELINE8	Stage 8 Fast Filter and Baseline Results	0x00000000	R
0x400840F0	CT_BASELINE9	Stage 9 Fast Filter and Baseline Results	0x00000000	R
0x400840F4	CT_BASELINE10	Stage 10 Fast Filter and Baseline Results	0x00000000	R
0x400840F8	CT_BASELINE11	Stage 11 Fast Filter and Baseline Results	0x00000000	R
0x400840FC	CT_BASELINE12	Stage 12 Fast Filter and Baseline Results	0x00000000	R
0x40084100	CT_BASELINE13	Stage 13 Fast Filter and Baseline Results	0x00000000	R
0x40084104	CT_BASELINE14	Stage 14 Fast Filter and Baseline Results	0x00000000	R
0x40084108	CT_BASELINE15	Stage 15 Fast Filter and Baseline Results	0x00000000	R
0x4008410C	CT_PK2PK0	Stage 0 and Stage 1 Peak-to-Peak Noise Results	0x00000000	R
0x40084110	CT_PK2PK1	Stage 2 and Stage 3 Peak-to-Peak Noise Results	0x00000000	R
0x40084114	CT_PK2PK2	Stage 4 and Stage 5 Peak-to-Peak Noise Results	0x00000000	R
0x40084118	CT_PK2PK3	Stage 6 and Stage 7 Peak-to-Peak Noise Results	0x00000000	R
0x4008411C	CT_PK2PK4	Stage 8 and Stage 9 Peak-to-Peak Noise Results	0x00000000	R
0x40084120	CT_PK2PK5	Stage 10 and Stage 11 Peak-to-Peak Noise Results	0x00000000	R
0x40084124	CT_PK2PK6	Stage 12 and Stage 13 Peak-to-Peak Noise Results	0x00000000	R
0x40084128	CT_PK2PK7	Stage 14 and Stage 15 Peak-to-Peak Noise Results	0x00000000	R

Power CDC Control Register

Address: 0x40084000, Reset: 0x00000000, Name: CT_CDC_PWR

Table 564. Bit Descriptions for CT_CDC_PWR

Bits	Bit Name	Description	Reset	Access
[31:11]	RESERVED	These bits read 0.	0x00000	R
[10:7]	UPD_COUNT	Readback proximity update count. Current count of the number of conversion cycles between fast filter updates.	0x0	R
6	STOP_BASELINE_UPD	Set to 1 to prevent updates to the CT_BASELINEx registers.	0x0	RW
5	PROX_SW	Force a proximity detection. Force proximity detection. Setting this bit forces a positive proximity detection, even if proximity is not actually detected. 0: do not force a positive proximity detection. 1: forces a positive proximity detection.	0x0	RW
4	RESERVED	Read 0. Set this bit to 1 while software is accessing the result registers.	0x0	R
3	SW_STATUS	Sequencer busy when high. CapTouch AFE status. 0: AFE is not currently converting. 1: AFE is currently converting.	0x0	R
1	SW_START_SEQ	Start a single sequence and measure all configured sensors. Software Initiates a preconfigured sequence. This bit is self clearing. 1: initiate a single preconfigured sequence.	0x0	RWAC
0	PWR_MODE	Force low power mode. Sets the operating modes. 0: full power mode. 1: full shutdown mode.	0x0	RW

CapTouch Control Configuration Register 1

Address: 0x40084004, Reset: 0x00000064, Name: CT_CFG1

Table 565. Bit Descriptions for CT_CFG1

Bits	Bit Name	Description	Reset	Access
[31:27]	RESERVED	These bits read 0.	0x00	R
[26:24]	INT_CYCLE	Number of capacitance (C) to voltage (V) integration cycles. 000: 1 cycles. 001: 2 cycles. 010: 4 cycles. 011: 8 cycles. 1xx: 16 cycles.	0x0	RW
[23:22]	AIN_SEL	Drive unmatched pads. 00: unmatched pads connected to AC_DRV. 01: unmatched pads connected to DC_DRV. 10: unmatched pads connected to GND. 11: unmatched pads floated.	0x0	RW
[15:13]	RESERVED	These bits read 0.	0x0	R
[12:10]	AUTOZERO_DUTY	Set C to V automatic zero duty cycle. Set high level period for the capacitance to voltage automatic zero signal. 000: 2 µs. 001: 3 µs. 010: 4 µs. … 111: 9 µs.	0x0	RW
9	RESERVED	These bits read 0.	0x0	R
8	BYPASS_GAIN	Override Bit 23 of CT_STAGEn. Bypass gain amplifier for all capacitance stages. Overrides the per stage AFE_GAIN bits in the STAGEn_CFG register. 0: do not override the stage PGA_BYPASS bit. 1: override the stage PGA_BYPASS bit.	0x0	RW
[7:6]	INT_BUFFER	Defaults to 3 µA. Internal buffer bias current control. 00: 2.5 µA. 01: 3 µA. 10: 3.5 µA. 11: 4 µA.	0x1	RW
[5:3]	PGA_BIAS	Defaults to 5 µA. Gain amplifier bias current control tool. 000: 3 µA. 001: 3.5 µA. 010: 4 µA. … 111: 6.5 µA.	0x4	RW
[2:0]	C2V_BIAS	Defaults to 10 µA. Capacitance to voltage bias current control. 000: 6 µA. 001: 7 µA. 010: 8 µA. … 111: 13 µA.	0x4	RW

CapTouch Control Configuration Register 2

Address: 0x40084008, Reset: 0x00000000, Name: CT_CFG2

Table 566. Bit Descriptions for CT_CFG2

Bits	Bit Name	Description	Reset	Access
31	C2V_LPF	Low pass filter on C to V summing nodes control. 0: low-pass filter off. 1: low-pass filter on.	0x0	RW
[30:28]	RESERVED	Read 0.	0x0	R
27	STIMER_EN	Enable self timed operation.	0x0	RW
[26:22]	C2V_HOLD_TIME	Set the hold time after Phase 4 has finished. 00000: 0.25 µs. 00001: 0.5 µs. … 10000: 4 µs. … 11111: 8 µs.	0x00	RW
[21:13]	SELF_TIMER_WAIT	Delay between sequences when the self timer is enabled. Set the delay between the end of a conversion set and the start of the next. During this time, the AFE is shut down, but all static registers are retained. This mode register must be used for low power standby mode or for lower power active modes where full speed is not required. Delay = CONV_SET_DELAY/512. 000000000: invalid. No measurements done. 000000001: 1.953 ms. 000000010: 3.906 ms. … 111111110: 996.1 ms. 111111111: 998.05 ms.	0x000	RW
[12:5]	PH24_DUTY	Sets the high level period of Phase 2 and Phase 4 for capacitance measurements. 00000000: 62.5 ns. 00000001: 125 ns. … 01000000: 4.0625 µs. … 01111111: 8 µs. 11111111: 16 µs.	0x00	RW
[4:0]	PH13_DUTY	Sets the high level period of Phase 1 and Phase 3 for capacitance measurements. 00000: 0.25 µs. 00001: 0.5 µs. … 10000: 4.25 µs. … 11111: 8 µs.	0x00	RW

AFE Control Configuration Register 3

Address: 0x4008400C, Reset: 0x00000000, Name: CT_CFG3

Table 567. Bit Descriptions for CT_CFG3

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	These bits read 0.	0x0	R
[27:25]	PK2PK_AVG	Number of samples to be averaged when calculating CDC value during peak-to-peak noise measurement stages. 000: 2 samples. 001: 4 samples. 010: 8 samples. 011: 16 samples. 100: 32 samples. Other settings: invalid.	0x0	RW
[24:20]	PK2PK_AVG_MIN	Minimum rank position for averaging. Minimum rank position for averager when calculating CDC value during peak-to-peak noise measurement stages.	0x00	RW
[19:16]	PK2PK_SUBSET	Noise = abs((#pk2pk_num_spls-1) − pk2pk_subset). Samples used for the peak-to-peak noise calculations. Calculation uses sample number pk2pk_subset and subtracts it from sample number (pk2pk_no-1 − pk2pk_subset) to calculate the peak-to-peak noise.	0x0	RW
[15:14]	PK2PK_NUM_SPLS	Number of samples for each peak-to-peak measurement. Number of samples for peak-to-peak noise calculation. 00: 4 samples. 01: 8 samples. 10: 16 samples. 11: 32 samples.	0x0	RW
13	RES_SEL	Result select. Selects what data is stored in the CDC_RESn registers. 0: CDC_RESn contains the filtered CDC result. 1: CDC_RESn contains the difference between the CDC result and the baseline value (BASELINEn).	0x0	RW
[12:11]	RESERVED	These bits read 0.	0x0	R
[10:5]	DIG_CTOV_CAPTOUCH_EN	Enable CapTouch operation: ensure that the GPCON bits are set correctly if setting these bits.	0x00	RW
[4:1]	IIR_WEIGHT	CDC IIR filter weight. Set the coefficient weight for the IIR filter.	0x0	RW
0	IIR_EN	Enable CDC IIR filter. Enable IIR filtering. 0: disable IIR filtering. 1: enable IIR filtering.	0x0	RW

Capacitance Measurement Stage Selection Register

Address: 0x40084010, Reset: 0x00000000, Name: CT_MEAS_SEL

Table 568. Bit Descriptions for CT_MEAS_SEL

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	These bits read 0.	0x0	R
[15:0]	GND_SEL	Enable selected stage(s). Enable measurement for measurement stages. Set Bit n = 1 to enable measurement on Stage n.	0x0	RW

Baseline Control Settings Register

Address: 0x40084014, Reset: 0x00000000, Name: CT_BASELINE_CTRL

Table 569. Bit Descriptions for CT_BASELINE_CTRL

Bits	Bit Name	Description	Reset	Access
[31:29]	BASELINE_CAL_DELAY	After proximity is cleared, the AFE waits BASELINE_CAL_DELAY before updating baseline values again. 000: 8× FAST_FILTER_UPDATE_RATE. 001: 16× FAST_FILTER_UPDATE_RATE. 010: 24× FAST_FILTER_UPDATE_RATE. 011: 32× FAST_FILTER_UPDATE_RATE. 100: 40× FAST_FILTER_UPDATE_RATE. 101: 48× FAST_FILTER_UPDATE_RATE. 110: 56× FAST_FILTER_UPDATE_RATE. 111: 64× FAST_FILTER_UPDATE_RATE.	0x0	RW
28	CONVERSION_RESET	Conversion reset. This bit is self clearing. 0: normal operation. 1: resets the conversion sequence to Capacitance Stage 0.	0x0	RWAC
27	FORCE_CAL	Force calibration during next sequence. Force calibration. This bit is self clearing. 0: normal operation. 1: force calibration of all stages.	0x0	RWAC
[26:24]	FAST_FILTER_UPDATE	Update rate for the baseline calculation fast filter. New samples are input to the fast filter based on FAST_FILTER_UPDATE. 000: new data input to fast filter every 2nd conversion cycle. 001: new data input to fast filter every 4th conversion cycle. 010: new data input to fast filter every 6th conversion cycle. 011: new data input to fast filter every 8th conversion cycle. 100: new data input to fast filter every 10th conversion cycle. 101: new data input to fast filter every 12th conversion cycle. 110: new data input to fast filter every 14th conversion cycle. 111: new data input to fast filter every 16th conversion cycle.	0x0	RW
[23:16]	FAST_PROX	If the difference between RESn and the average CDC result is greater than (FAST_PROX × 4), fast proximity is detected.	0x0	RW
[15:8]	SLOW_PROX	If the difference between BASELINE and the average CDC result is greater than (SLOW_PROX × 4), slow proximity is detected.	0x0	RW
[7:4]	BL_COEFF	Baseline calculation slow filter new sample weight. Sample weight = BL_COEFF + 1. 0000: 1. 0001: 2. … 1111: 16.	0x0	RW
[3:0]	CDC_COEFF	Baseline calculation fast filter new sample weight. Sample weight = CDC_COEFF + 1. 0000: 1. 0001: 2. … 1111: 16.	0x0	RW

Setup of the Rank-and-Average Filtering Register

Address: 0x40084018, Reset: 0x00000000, Name: CT_AVG

Table 570. Bit Descriptions for CT_AVG

Bits	Bit Name	Description	Reset	Access
31	RESERVED	These bits read 0.	0x0	R
[30:28]	RANK_FILT	Number of samples to be ranked. 000: no ranking (samples are just stored). Number of samples to be gathered is selected by AVG_GROUP1 bits. 001: 4 samples. 010: 8 samples. 011: 16 samples. 100: 32 samples.	0x0	RW
[27:26]	RESERVED	These bits read 0.	0x0	R
[25:24]	AVG_OUTPUT	Selects the final averaged value. 00: no averaging (last value in from rank-filter is output). 01: average of Group 1. 10: average of Group 2. 11: average of (average of Group 1 and average of Group 2).	0x0	RW
23	RESERVED	These bits read 0.	0x0	R
[22:20]	AVG_GROUP2	Number of samples in Averager Group 2. 000: no samples (or one sample if RANK_FILT = 000). 001: 2 samples. 010: 4 samples. 011: 8 samples. 100: 16 samples. 101: 32 samples. Other settings: not used.	0x0	RW
19	RESERVED	These bits read 0.	0x0	R
[18:16]	AVG_GROUP1	Number of samples in Averager Group 1. 000: no samples (or 1 sample if RANK_FILT = 000). 001: 2 samples. 010: 4 samples. 011: 8 samples. 100: 16 samples. 101: 32 samples. Other settings: not used.	0x0	RW
[15:13]	RESERVED	These bits read 0.	0x0	R
[12:8]	AVG_MIN_GROUP2	Minimum rank position to be averaged in Averager Set 2. If RANK_FILT = 000, the upper rank filter samples are the most recent samples.	0x0	RW
[7:5]	RESERVED	These bits read 0.	0x0	R
[4:0]	AVG_MIN_GROUP1	Minimum rank position to be averaged in Averager Set 1. If RANK_FILT = 000, the upper rank filter samples are the most recent samples.	0x0	RW

Enable Calibration for Measurement Stages Register

Address: 0x4008401C, Reset: 0x00000000, Name: CT_CAL_EN

Table 571. Bit Descriptions for CT_CAL_EN

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	These bits read 0.	0x0	R
[15:0]	CAL_EN	Enable calibration for measurement stages. Set Bit n = 1 to enable calibration on Stage n.	0x0000	RW

Touch Detection Thresholds Register

Address: 0x40084020, Reset: 0x00000000, Name: CT_TOUCH_CFG1

Table 572. Bit Descriptions for CT_TOUCH_CFG1

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	Reserved.	0x0	R
[29:16]	TCH_UPPER_THLD	Sensor upper touch threshold.	0x0	RW
[15:14]	RESERVED	Reserved.	0x0	R
[13:0]	TCH_LOWER_THLD	Sensor lower touch threshold.	0x0	RW

Touch Detection Timeout Register

Address: 0x40084024, Reset: 0x00000032, Name: CT_TOUCH_CFG2

Table 573. Bit Descriptions for CT_TOUCH_CFG2

Bits	Bit Name	Description	Reset	Access
[31:24]	RESERVED	These bits read 0.	0x00	R
23	INDIVIDUAL_THRESHOLD_EN	Use CT_SENSOR_THR_CFGx for sensor threshold.	0x0	RW
22	TCH_EVENT_ON_HIGH	Detect rising edge when set. 0: active if Input Source A − Input Source B < threshold for minimum duration. 1: active if Input Source A − Input Source B > threshold for minimum duration.	0x0	RW
[21:19]	TCH_SEL_INPA	Select Input Source A. 000: load all zeros. 001: load CDC value. 010: load slow filter output. 011: load fast filter output. 100: load ambient value.	0x0	RW
[18:16]	TCH_SEL_INPB	Select input Source B. 000: load all zeros. 001: load CDC value. 010: load slow filter output. 011: load fast filter output. 100: load ambient value.	0x0	RW
15	TCH_DETECT_ENABLE	Enable touch detector.	0x0	RW
[14:10]	RESERVED	These bits read 0.	0x00	R
[9:0]	TCH_MIN_DURATION	Minimum touch duration (minimum duration = 2).	0x032	RW

Release Detection Thresholds Register

Address: 0x40084028, Reset: 0x00000000, Name: CT_RELEASE_CFG1

Table 574. Bit Descriptions for CT_RELEASE_CFG1

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	Reserved.	0x0	R
[29:16]	REL_UPPER_THLD	Sensor upper release threshold.	0x0	RW
[15:14]	RESERVED	Reserved.	0x0	R
[13:0]	REL_LOWER_THLD	Sensor lower release threshold.	0x0	RW

Release Detection Timeout Register

Address: 0x4008402C, Reset: 0x00000032, Name: CT_RELEASE_CFG2

Table 575. Bit Descriptions for CT_RELEASE_CFG2

Bits	Bit Name	Description	Reset	Access
[31:24]	RESERVED	These bits read 0.	0x00	R
23	INDIVIDUAL_THRESHOLD_EN	Use CT_SENSOR_THR_CFGx for sensor threshold.	0x0	RW
22	REL_EVENT_ON_HIGH	Determine calculation direction. 0: active if Input Source A − Input Source B < threshold for minimum duration. 1: active if Input Source A − Input Source B > threshold for minimum duration.	0x0	RW
[21:19]	REL_SEL_INPA	Select Input Source A. 000: load all zeros. 001: load CDC value. 010: load slow filter output. 011: load fast filter output. 100: load ambient value.	0x0	RW
[18:16]	REL_SEL_INPB	Select Input Source B. 000: load all zeros. 001: load CDC value. 010: load slow filter output. 011: load fast filter output. 100: load ambient value.	0x0	RW
15	REL_DETECT_ENABLE	Enable the release detector.	0x0	RW
[14:10]	RESERVED	Read 0.	0x00	R
[9:0]	REL_MIN_DURATION	Minimum duration before release is detected (minimum duration = 2).	0x032	RW

Interrupt Enable Register

Address: 0x40084030, Reset: 0x00000000, Name: CT_IEN

Table 576. Bit Descriptions for CT_IEN

Bits	Bit Name	Description	Reset	Access
[31:5]	RESERVED	These bits read 0.	0x0	R
4	RELEASE_DETECTED_IEN	Release detect interrupt enable. RELEASE_DETECTED interrupt enable.	0x0	RW
3	TOUCH_DETECTED_IEN	Touch detect interrupt enable. TOUCH_DETECTED interrupt enable.	0x0	RW
2	PROX_DETECTED_IEN	Proximity detect interrupt enable. PROX_DETECTED interrupt enable.	0x0	RW
1	STATUS_GT_THRESHOLD_IEN	LSTAT or HSTAT has exceeded threshold interrupt enable. BASELINE_UPD interrupt enable.	0x0	RW
0	CONV_SET_COMPLETE_IEN	Sequence complete interrupt enable. CONV_SET_COMPLETE interrupt enable.	0x0	RW

Primary Interrupt Register

Address: 0x40084034, Reset: 0x00000000, Name: CT_INT

Primary interrupt register. The bits in this register are sticky when set. Each bit is cleared by writing a 1 to its location. Writing a 0 has no effect. If simultaneously the interrupt source is asserted and the core is attempting to clear a bit, the interrupt remains set.

A 1 means the interrupt source has been asserted since the last time the bit was cleared. A 0 means the interrupt source has not been asserted since the last time the bit was cleared.

Table 577. Bit Descriptions for CT_INT

Bits	Bit Name	Description	Reset	Access
[31:5]	RESERVED	These bits read 0.	0x0	R
4	RELEASE_DETECTED	Sticky bit: write 1 to clear. Offset limit exceeded in negative direction on one of the sampled buttons. Poll OFFS_LSTAT to determine which one.	0x0	RW1C
3	TOUCH_DETECTED	Sticky bit: write 1 to clear. Offset limit exceeded in positive direction on one of the sampled buttons. Poll OFFS_HSTAT to determine which one.	0x0	RW1C
2	PROX_DETECTED	Sticky bit: write 1 to clear. Proximity event detected interrupt.	0x0	RW1C
1	STATUS_GT_THRESHOLD	Sticky bit: write 1 to clear. Baseline value updated interrupt.	0x0	RW1C
0	CONV_SET_COMPLETE	Sticky bit: write 1 to clear. Conversion set completed interrupt.	0x0	RW1C

Stage Offset High Interrupt Status Register

Address: 0x40084038, Reset: 0x00000000, Name: CT_OFFS_HSTAT

These bits are read only, they reflect the current high limit status for each stage.

Table 578. Bit Descriptions for CT_OFFS_HSTAT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	These bits read 0.	0x0	R
[15:0]	OFFS_HSTAT	Stage n has exceeded the high limit threshold. Each Bit n indicates the high limit status of Stage n. 1: (CDC_n − BASELINEn) > SENSOR_THRESHOLDn.	0x0	R

Stage Offset Low Interrupt Status Register

Address: 0x4008403C, Reset: 0x00000000, Name: CT_OFFS_LSTAT

These bits are read only, they reflect the current low-limit status for each stage.

Table 579. Bit Descriptions for CT_OFFS_LSTAT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	These bits read 0.	0x0	R
[15:0]	OFFS_LSTAT	Stage_n has exceeded the low limit threshold. Each Bit n indicates the low limit status of Stage n. Bit n = 1: (BASELINEn − CDC_VALn) ≥ SENSOR_THRESHOLDn.	0x0	R

Stage Proximity Status Register

Address: 0x40084040, Reset: 0x00000000, Name: CT_PROX_STAT

Table 580. Bit Descriptions for CT_PROX_STAT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	These bits read 0.	0x0	R
[15:0]	PROX_STAT	Proximity status per stage. Each Bit n indicates the proximity status of Stage n. Bit n = 0: proximity detected on Stage n. Bit n = 1: proximity detected on Stage n.	0x0	R

Stage Fast Proximity Status Register

Address: 0x40084044, Reset: 0x00000000, Name: CT_FPROX_STAT

Table 581. Bit Descriptions for CT_FPROX_STAT

Bits	Bit Name	Description	Reset	Access
[31:16]	RESERVED	These bits read 0.	0x0	R
[15:0]	FPROX_STAT	Fast proximity status per stage. Each Bit n indicates the fast proximity status of Stage n. Bit n = 0: fast proximity not detected on Stage n. Bit n = 1: fast proximity detected on Stage n.	0x0	R

Stage Touch/Release Status Register

Address: 0x40084048, Reset: 0x00000000, Name: CT_TOUCH_STAT

Table 582. Bit Descriptions for CT_TOUCH_STAT

Bits	Bit Name	Description	Reset	Access
[31:16]	REL_DETECT_STAT	Sticky. Release detect status per stage. 0: release not detected on Stage n. 1: release detected on Stage n.	0x0000	RW1C
[15:0]	TCH_DETECT_STAT	Sticky. Touch detect status per stage. Each Bit n indicates the fast proximity status of Stage n. Bit n = 0: touch not detected on Stage n. Bit n = 1: touch detected on Stage n.	0x0000	RW1C

Stage 0 Configuration Register

Address: 0x4008404C, Reset: 0x00000000, Name: CT_STAGE0_CFG

Table 583. Bit Descriptions for CT_STAGE0_CFG

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved.		RW
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor = ±9.324 pF.	0x0	RW

Bits	Bit Name	Description	Reset	Access
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset. 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 1 Configuration Register

Address: 0x40084050, Reset: 0x00000000, Name: CT_STAGE1_CFG

Table 584. Bit Descriptions for CT_STAGE1_CFG

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved.		RW
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ± 8.741 pF. 1111: maximum change in CIN capacitor = ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable Offset DAC. Enable MAIN-DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R

Bits	Bit Name	Description	Reset	Access
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset. 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 2 Configuration Register

Address: 0x40084054, Reset: 0x00000000, Name: CT_STAGE2_CFG

Table 585. Bit Descriptions for CT_STAGE2_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor = ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable main DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 3 Configuration Register

Address: 0x40084058, Reset: 0x00000000, Name: CT_STAGE3_CFG

Table 586. Bit Descriptions for CT_STAGE3_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor = ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 4 Configuration Register

Address: 0x4008405C, Reset: 0x00000000, Name: CT_STAGE4_CFG

Table 587. Bit Descriptions for CT_STAGE4_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency-hopping noise-reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset. 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 5 Configuration Register

Address: 0x40084060, Reset: 0x00000000, Name: CT_STAGE5_CFG

Table 588. Bit Descriptions for CT_STAGE5_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor = ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable Offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 6 Configuration Register

Address: 0x40084064, Reset: 0x00000000, Name: CT_STAGE6_CFG

Table 589. Bit Descriptions for CT_STAGE6_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 7 Configuration Register

Address: 0x40084068, Reset: 0x00000000, Name: CT_STAGE7_CFG

Table 590. Bit Descriptions for CT_STAGE7_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 8 Configuration Register

Address: 0x4008406C, Reset: 0x00000000, Name: CT_STAGE8_CFG

Table 591. Bit Descriptions for CT_STAGE8_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 9 Configuration Register

Address: 0x40084070, Reset: 0x00000000, Name: CT_STAGE9_CFG

Table 592. Bit Descriptions for CT_STAGE9_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 10 Configuration Register

Address: 0x40084074, Reset: 0x00000000, Name: CT_STAGE10_CFG

Table 593. Bit Descriptions for CT_STAGE10_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 11 Configuration Register

Address: 0x40084078, Reset: 0x00000000, Name: CT_STAGE11_CFG

Table 594. Bit Descriptions for CT_STAGE11_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 12 Configuration Register

Address: 0x4008407C, Reset: 0x00000000, Name: CT_STAGE12_CFG

Table 595. Bit Descriptions for CT_STAGE12_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0. Enable SUB-DAC.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 13 Configuration Register

Address: 0x40084080, Reset: 0x00000000, Name: CT_STAGE13_CFG

Table 596. Bit Descriptions for CT_STAGE13_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 14 Configuration Register

Address: 0x40084084, Reset: 0x00000000, Name: CT_STAGE14_CFG

Table 597. Bit Descriptions for CT_STAGE14_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0. Enable SUB_DAC.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 15 Configuration Register

Address: 0x40084088, Reset: 0x00000000, Name: CT_STAGE15_CFG

Table 598. Bit Descriptions for CT_STAGE15_CFG

Bits	Bit Name	Description	Reset	Access
27	CIN_CON_POS_CDC_EN	Enable selected CIN to positive CDC.	0x0	RW
[26:24]	CIN_CON_POS_CDC	Select CIN to connect to positive CDC input. Stage 0 connections setup. 000: BU_ONOFF. 001: BU_UP. 010: BU_DOWN. 011: BU_LEFT. 100: BU_RIGHT. 101: BU_ENTER.	0x0	RW
23	PGA_BYPASS	Bypass PGA for this stage. Gain bypass. If the BYPASS_GAIN bit in the CFG1 register is set to 1, this bit has no effect. 0: the gain for this stage is set by the PGA_GAIN bits. 1: the gain for this stage is bypassed.	0x0	RW
22	NOISE_REDUCTION_EN	Enable frequency hopping noise reduction.	0x0	RW
[21:20]	PGA_GAIN	Stage specific PGA gain setup. AFE gain adjust for Stage 0. 00: gain of 2. 01: gain of 4. 10: gain of 8. 11: gain of 16.	0x0	RW
[19:16]	C2V_IP_RANGE	0000: maximum change in CIN capacitor = ±0.58275 pF. 0001: maximum change in CIN capacitor = ±1.1655 pF. 0010: maximum change in CIN capacitor = ±1.7482 pF. … 1110: maximum change in CIN capacitor = ±8.741 pF. 1111: maximum change in CIN capacitor= ±9.324 pF.	0x0	RW
15	EN_MAIN_DAC	Enable offset DAC. Enable MAIN_DAC. 0: disable MAIN_DAC. 1: enable MAIN_DAC.	0x0	RW
14	RESERVED	Read 0.	0x0	R
13	PK2PK	Enable peak-to-peak measurements. Set this bit to perform peak-to-peak noise measurements for Stage 0. 0: disable peak-to-peak noise measurement for Stage 0. 1: enable peak-to-peak noise measurement for Stage 0.	0x0	RW
[12:10]	RESERVED	Read 0.	0x0	R
[9:3]	MAIN_DAC	Main DAC offset. 1 LSB = 2 × 388.5 fF. AFE MAIN_DAC offset setting for Stage 0. 1 LSB = 388.5 fF.	0x00	RW
[2:0]	SUB_DAC	Sub DAC offset 1 LSB = 2 × 48 fF. AFE SUB_DAC offset settings for Stage 0. 1 LSB = 48.5 fF.	0x0	RW

Stage 0 and Stage 1 Sensor Threshold Register

Address: 0x4008408C, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG0

Stage 0 and Stage 1 sensor threshold configuration.

Table 599. Bit Descriptions for CT_SENSOR_THR_CFG0

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD1	Capacitance Stage 1 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD0	Capacitance Stage 0 sensor threshold.	0x0	RW

Stage 2 and Stage 3 Sensor Threshold Register

Address: 0x40084090, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG1

Stage 0 and Stage 1 sensor threshold configuration.

Table 600. Bit Descriptions for CT_SENSOR_THR_CFG1

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD3	Capacitance Stage 3 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD2	Capacitance Stage 2 sensor threshold.	0x0	RW

Stage 4 and Stage 5 Sensor Threshold Register

Address: 0x40084094, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG2

Stage 0 and Stage 1 sensor threshold configuration.

Table 601. Bit Descriptions for CT_SENSOR_THR_CFG2

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD5	Capacitance Stage 5 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD4	Capacitance Stage 2 sensor threshold.	0x0	RW

Stage 6 and Stage 7 Sensor Threshold Register

Address: 0x40084098, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG3

Stage 0 and Stage 1 sensor threshold configuration.

Table 602. Bit Descriptions for CT_SENSOR_THR_CFG3

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD7	Capacitance Stage 7 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD6	Capacitance Stage 3 sensor threshold.	0x0	RW

Stage 8 and Stage 9 Sensor Threshold Register

Address: 0x4008409C, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG4

Stage 0 and Stage 1 sensor threshold configuration.

Table 603. Bit Descriptions for CT_SENSOR_THR_CFG4

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD9	Capacitance Stage 9 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD8	Capacitance Stage 8 sensor threshold.	0x0	RW

Stage 10 and Stage 11 Sensor Threshold Register

Address: 0x400840A0, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG5

Stage 0 and Stage 1 sensor threshold configuration.

Table 604. Bit Descriptions for CT_SENSOR_THR_CFG5

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD11	Capacitance Stage 11 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD10	Capacitance Stage 10 sensor threshold.	0x0	RW

Stage 12 and Stage 13 Sensor Threshold Register

Address: 0x400840A4, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG6

Stage 0 and Stage 1 sensor threshold configuration.

Table 605. Bit Descriptions for CT_SENSOR_THR_CFG6

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD13	Capacitance Stage 13 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD12	Capacitance Stage 12 sensor threshold.	0x0	RW

Stage 14 and Stage 15 Sensor Threshold Register

Address: 0x400840A8, Reset: 0x00000000, Name: CT_SENSOR_THR_CFG7

Stage 0 and Stage 1 sensor threshold configuration.

Table 606. Bit Descriptions for CT_SENSOR_THR_CFG7

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	SENSOR_THRESHOLD15	Capacitance Stage 15 sensor threshold.	0x0	RW
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	SENSOR_THRESHOLD14	Capacitance Stage 14 sensor threshold.	0x0	RW

Stage 0 and Stage 1 Results Register

Address: 0x400840AC, Reset: 0x00000000, Name: CT_CDC_RES0

Table 607. Bit Descriptions for CT_CDC_RES0

Bits	Bit Name	Description	Reset	Access
31	SIGN1	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE1|. 0: CDC result > BASELINE1. 1: CDC result < BASELINE1.	0x0	R
30	RESERVED	Read 0.	0x0	R
[29:16]	CDC_RES1	If RES_SEL = 0, the result from the Stage 1 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE1| is stored in these bits.	0x0000	R
15	SIGN0	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE0|. 0: CDC result > BASELINE0. 1: CDC result < BASELINE0.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES0	If RES_SEL = 0, the result from the Stage 0 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE0| is stored in these bits.	0x0000	R

Stage 2 and Stage 3 Results Register

Address: 0x400840B0, Reset: 0x00000000, Name: CT_CDC_RES1

Table 608. Bit Descriptions for CT_CDC_RES1

Bits	Bit Name	Description	Reset	Access
31	SIGN3	When RES_SEL = 1, this bit indicates the sign of the value |CDC result − BASELINE3|. 0: CDC result > BASELINE3. 1: CDC result < BASELINE3.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES3	If RES_SEL = 0, the result from the Stage 3 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result − BASELINE3| is stored in these bits.	0x0000	R
15	SIGN2	When RES_SEL = 1, this bit indicates the sign of the value |CDC result − BASELINE2|. 0: CDC result > BASELINE2. 1: CDC result < BASELINE2.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES2	If RES_SEL = 0, the result from the Stage 2 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result − BASELINE2| is stored in these bits.	0x0000	R

Stage 4 and Stage 5 Results Register

Address: 0x400840B4, Reset: 0x00000000, Name: CT_CDC_RES2

Table 609. Bit Descriptions for CT_CDC_RES2

Bits	Bit Name	Description	Reset	Access
31	SIGN5	When RES_SEL = 1, this bit indicates the sign of the value |CDC result − BASELINE5|. 0: CDC result > BASELINE5. 1: CDC result < BASELINE5.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES5	If RES_SEL = 0, the result from the Stage 5 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result − BASELINE5| is stored in these bits.	0x0000	R
15	SIGN4	When RES_SEL = 1, this bit indicates the sign of the value |CDC result − BASELINE4|. 0: CDC result > BASELINE4. 1: CDC result < BASELINE4.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES4	If RES_SEL = 0, the result from the Stage 4 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result − BASELINE4| is stored in these bits.	0x0000	R

Stage 6 and Stage 7 Results Register

Address: 0x400840B8, Reset: 0x00000000, Name: CT_CDC_RES3

Table 610. Bit Descriptions for CT_CDC_RES3

Bits	Bit Name	Description	Reset	Access
31	SIGN7	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE7|. 0: CDC result > BASELINE7. 1: CDC result < BASELINE7.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES7	If RES_SEL = 0, the result from the Stage 7 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE7| is stored in these bits.	0x0000	R
15	SIGN6	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE6|. 0: CDC result > BASELINE6. 1: CDC result < BASELINE6.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES6	If RES_SEL = 0, the result from the Stage 6 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE6| is stored in these bits.	0x0000	R

Stage 8 and Stage 9 Results Register

Address: 0x400840BC, Reset: 0x00000000, Name: CT_CDC_RES4

Table 611. Bit Descriptions for CT_CDC_RES4

Bits	Bit Name	Description	Reset	Access
31	SIGN9	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE9|. 0: CDC result > BASELINE9. 1: CDC result < BASELINE9.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES9	If RES_SEL = 0, the result from the Stage 9 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE9| is stored in these bits.	0x0000	R
15	SIGN8	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE8|. 0: CDC result > BASELINE8. 1: CDC result < BASELINE8.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES8	If RES_SEL = 0, the result from the Stage 8 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE8| is stored in these bits.	0x0000	R

Stage 10 and Stage 11 Results Register

Address: 0x400840C0, Reset: 0x00000000, Name: CT_CDC_RES5

Table 612. Bit Descriptions for CT_CDC_RES5

Bits	Bit Name	Description	Reset	Access
31	SIGN11	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE11|. 0: CDC result > BASELINE11. 1: CDC result < BASELINE11.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES11	If RES_SEL = 0, the result from the Stage 11 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE11| is stored in these bits.	0x0000	R
15	SIGN10	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE10|. 0: CDC result > BASELINE10. 1: CDC result < BASELINE10.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES10	If RES_SEL = 0, the result from the Stage 10 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE10| is stored in these bits.	0x0000	R

Stage 12 and Stage 13 Results Register

Address: 0x400840C4, Reset: 0x00000000, Name: CT_CDC_RES6

Table 613. Bit Descriptions for CT_CDC_RES6

Bits	Bit Name	Description	Reset	Access
31	SIGN13	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE13|. 0: CDC result > BASELINE13. 1: CDC result < BASELINE13.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES13	If RES_SEL = 0, the result from the Stage 13 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE13| is stored in these bits.	0x0000	R
15	SIGN12	When RES_SEL = 1, this bit indicates the sign of the value |CDC result – BASELINE12|. 0: CDC result > BASELINE12. 1: CDC result < BASELINE12.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES12	If RES_SEL = 0, the result from the Stage 12 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result – BASELINE12| is stored in these bits.	0x0000	R

Stage 14 and Stage 15 Results Register

Address: 0x400840C8, Reset: 0x00000000, Name: CT_CDC_RES7

Table 614. Bit Descriptions for CT_CDC_RES7

Bits	Bit Name	Description	Reset	Access
31	SIGN15	When RES_SEL = 1, this bit indicates the sign of the value |CDC result − BASELINE15|. 0: CDC result > BASELINE15. 1: CDC result < BASELINE15.	0x0	R
30	RESERVED	This bit reads 0.	0x0	R
[29:16]	CDC_RES15	If RES_SEL = 0, the result from the Stage 15 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result − BASELINE15| is stored in these bits.	0x0000	R
15	SIGN14	When RES_SEL = 1, this bit indicates the sign of the value |CDC result − BASELINE14|. 0: CDC result > BASELINE14. 1: CDC result < BASELINE14.	0x0	R
14	RESERVED	This bit reads 0.	0x0	R
[13:0]	CDC_RES14	If RES_SEL = 0, the result from the Stage 14 capacitance CDC conversion is stored in these bits. If RES_SEL = 1, |CDC result − BASELINE14| is stored in these bits.	0x0000	R

Stage 0 Fast Filter and Baseline Results Register

Address: 0x400840CC, Reset: 0x00000000, Name: CT_BASELINE0

Table 615. Bit Descriptions for CT_BASELINE0

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG0	Stage 0 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE0	Stage 0 baseline result.	0x0	R

Stage 1 Fast Filter and Baseline Results Register

Address: 0x400840D0, Reset: 0x00000000, Name: CT_BASELINE1

Table 616. Bit Descriptions for CT_BASELINE1

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG1	Stage 1 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE1	Stage 1 baseline result.	0x0	R

Stage 2 Fast Filter and Baseline Results Register

Address: 0x400840D4, Reset: 0x00000000, Name: CT_BASELINE2

Table 617. Bit Descriptions for CT_BASELINE2

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG2	Stage 2 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE2	Stage 2 baseline result.	0x0	R

Stage 3 Fast Filter and Baseline Results Register

Address: 0x400840D8, Reset: 0x00000000, Name: CT_BASELINE3

Table 618. Bit Descriptions for CT_BASELINE3

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG3	Stage 3 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE3	Stage 3 baseline result.	0x0	R

Stage 4 Fast Filter and Baseline Results Register

Address: 0x400840DC, Reset: 0x00000000, Name: CT_BASELINE4

Table 619. Bit Descriptions for CT_BASELINE4

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG4	Stage 4 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE4	Stage 4 baseline result.	0x0	R

Stage 5 Fast Filter and Baseline Results Register

Address: 0x400840E0, Reset: 0x00000000, Name: CT_BASELINE5

Table 620. Bit Descriptions for CT_BASELINE5

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG5	Stage 5 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE5	Stage 5 baseline result.	0x0	R

Stage 6 Fast Filter and Baseline Results Register

Address: 0x400840E4, Reset: 0x00000000, Name: CT_BASELINE6

Table 621. Bit Descriptions for CT_BASELINE6

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG6	Stage 6 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE6	Stage 6 baseline result.	0x0	R

Stage 7 Fast Filter and Baseline Results Register

Address: 0x400840E8, Reset: 0x00000000, Name: CT_BASELINE7

Table 622. Bit Descriptions for CT_BASELINE7

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG7	Stage 7 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE7	Stage 7 baseline result.	0x0	R

Stage 8 Fast Filter and Baseline Results Register

Address: 0x400840EC, Reset: 0x00000000, Name: CT_BASELINE8

Table 623. Bit Descriptions for CT_BASELINE8

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG8	Stage 8 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE8	Stage 8 baseline result.	0x0	R

Stage 9 Fast Filter and Baseline Results Register

Address: 0x400840F0, Reset: 0x00000000, Name: CT_BASELINE9

Table 624. Bit Descriptions for CT_BASELINE9

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG9	Stage 9 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE9	Stage 9 baseline result.	0x0	R

Stage 10 Fast Filter and Baseline Results Register

Address: 0x400840F4, Reset: 0x00000000, Name: CT_BASELINE10

Table 625. Bit Descriptions for CT_BASELINE10

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG10	Stage 10 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE10	Stage 10 baseline result.	0x0	R

Stage 11 Fast Filter and Baseline Results Register

Address: 0x400840F8, Reset: 0x00000000, Name: CT_BASELINE11

Table 626. Bit Descriptions for CT_BASELINE11

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG11	Stage 11 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE11	Stage 11 baseline result.	0x0	R

Stage 12 Fast Filter and Baseline Results Register

Address: 0x400840FC, Reset: 0x00000000, Name: CT_BASELINE12

Table 627. Bit Descriptions for CT_BASELINE12

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG12	Stage 12 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE12	Stage 12 baseline result.	0x0	R

Stage 13 Fast Filter and Baseline Results Register

Address: 0x40084100, Reset: 0x00000000, Name: CT_BASELINE13

Table 628. Bit Descriptions for CT_BASELINE13

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG13	Stage 13 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE13	Stage 13 baseline result.	0x0	R

Stage 14 Fast Filter and Baseline Results Register

Address: 0x40084104, Reset: 0x00000000, Name: CT_BASELINE14

Table 629. Bit Descriptions for CT_BASELINE14

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG14	Stage 14 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE14	Stage 14 baseline result.	0x0	R

Stage 15 Fast Filter and Baseline Results Register

Address: 0x40084108, Reset: 0x00000000, Name: CT_BASELINE15

Table 630. Bit Descriptions for CT_BASELINE15

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	CDC_AVG15	Stage 15 fast filter results.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	BASELINE15	Stage 15 baseline result.	0x0	R

Stage 0 and Stage 1 Peak-to-Peak Noise Results Register

Address: 0x4008410C, Reset: 0x00000000, Name: CT_PK2PK0

Table 631. Bit Descriptions for CT_PK2PK0

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK1	Stage 1 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK0	Stage 0 peak-to-peak noise measurement result.	0x0	R

Stage 2 and Stage 3 Peak-to-Peak Noise Results Register

Address: 0x40084110, Reset: 0x00000000, Name: CT_PK2PK1

Table 632. Bit Descriptions for CT_PK2PK1

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK3	Stage 3 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK2	Stage 2 peak-to-peak noise measurement result.	0x0	R

Stage 4 and Stage 5 Peak-to-Peak Noise Results Register

Address: 0x40084114, Reset: 0x00000000, Name: CT_PK2PK2

Table 633. Bit Descriptions for CT_PK2PK2

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK5	Stage 5 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK4	Stage 4 peak-to-peak noise measurement result.	0x0	R

Stage 6 and Stage 7 Peak-to-Peak Noise Results Register

Address: 0x40084118, Reset: 0x00000000, Name: CT_PK2PK3

Table 634. Bit Descriptions for CT_PK2PK3

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK7	Stage 7 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK6	Stage 6 peak-to-peak noise measurement result.	0x0	R

Stage 8 and Stage 9 Peak-to-Peak Noise Results Register

Address: 0x4008411C, Reset: 0x00000000, Name: CT_PK2PK4

Table 635. Bit Descriptions for CT_PK2PK4

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK9	Stage 9 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK8	Stage 8 peak-to-peak noise measurement result.	0x0	R

Stage 10 and Stage 11 Peak-to-Peak Noise Results Register

Address: 0x40084120, Reset: 0x00000000, Name: CT_PK2PK5

Table 636. Bit Descriptions for CT_PK2PK5

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK11	Stage 11 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK10	Stage 10 peak-to-peak noise measurement result.	0x0	R

Stage 12 and Stage 13 Peak-to-Peak Noise Results Register

Address: 0x40084124, Reset: 0x00000000, Name: CT_PK2PK6

Table 637. Bit Descriptions for CT_PK2PK6

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	These bits read 0.	0x0	R
[29:16]	PK2PK13	Stage 13 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	These bits read 0.	0x0	R
[13:0]	PK2PK12	Stage 12 peak-to-peak noise measurement result.	0x0	R

Stage 14 and Stage 15 Peak-to-Peak Noise Results Register

Address: 0x40084128, Reset: 0x00000000, Name: CT_PK2PK7

Table 638. Bit Descriptions for CT_PK2PK7

Bits	Bit Name	Description	Reset	Access
[31:30]	RESERVED	Read 0.	0x0	R
[29:16]	PK2PK15	Stage 15 peak-to-peak noise measurement result.	0x0	R
[15:14]	RESERVED	Read 0.	0x0	R
[13:0]	PK2PK14	Stage 14 peak-to-peak noise measurement result.	0x0	R

PINS

INTRODUCTION

The ADuCM350 is packaged in a 120-ball chip scale package ball grid array (CSP_BGA).

The pin function description table (Table 639) for the package is split into the following sections:

- Power and grounds
- AFE pins
- Debug interface
- Serial Port H
- Other serial ports
- USB
- CapTouch
- Crystals
- Display
- Miscellaneous digital input/output
- Audio

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	DNC	P2.1/ COM1/ RESX	P2.3/ COM3/ DCX	P2.5/S2/ ECLOCK-WRX	P1.0/S3/ D0/SCL	P1.2/S5/ D2/DIN	P1.4/S7/ D4	P1.6/S9/ D6	P1.8/S11/ D8	P1.10/ S13/D10	P1.12/ S15/D12	P1.14/ S17/D14	P2.10/S23	P2.9/S22	P2.15/S28
B	V_LCD_23	P2.0/ COM0	P2.2/ COM2/ CSX	P2.4/S1/ RWX-RDX	P1.1/S4/ D1/DOUT	P1.3/S6/ D3	P1.5/S8/ D5	P1.7/S10/ D7	P1.9/S12/ D9	P1.11/ S14/D11	P1.13/ S16/D13	P1.15/ S18/D15	P2.11/S24	P2.12/S25	P2.8/S21
C	HF_XTAL2	V_LCD_13												P3.10/S31	P2.7/S20/ TOUTA
D	HF_XTAL1	VLCDVDD												P2.13/S26	P2.6/ S19/TE
E	VUSB	VLCD FLY1												P3.11/S32	P2.14/S27
F	USB DM	VLCD FLY2				DGND USB	DGND2/ LCD_GND	P3.8/S29	P3.9/S30	P3.3/ SPI0_CS				P3.0/ SPIO_SCLK	P3.2/ SPIO_MOSI
G	USB DP	VBUS				DGND1				P3.4/ I2CSCL/ SPI1_SCLK				P3.1/ SPIO_MISO	P3.6/UTX/ TOUTB/ SPI1_MOSI
H	TMS-SWDIO/ P0.8	TCK-SWCLK/ P0.9				VCCM_ DIG				P3.5/ I2CSD/ SPI1_MISO				DGND	VDD_IO
J	TDO-SWO/ P0.6/UTX	TDI/P0.7/ URX				P3.14/ LRCLK				P3.7/URX/ TOUTC/ SPI1_CS				P0.4/ CAPT_E	P0.1/ CAPT_B
K	DVDD	P0.11				P3.12/ BEEP/ BMCLK	P3.13/ BEEPX/ SDATA	RESETX	AGND_RX/ AGND_TX	AGND_REF				P0.3/ CAPT_D	P0.0/ CAPT_A
L	P4.0/ I2CSCL	P4.1/ I2CSD												P0.5/ CAPT_F	P0.2/ CAPT_C
M	VBACK	P0.15/ SPIH_CS												TRACE0	TRACECLK
N	LF_XTAL2	BOOT												TRACE2	TRACE1
P	LF_XTAL1	P0.13/ SPIH_MISO	P0.14/ SPIH_MOSI	VCCM_ ANA	RCAL 1	AFE 2	AFE 4	AFE 6	AFE 8	VBIAS	TIA_I	AN_A	AN_B	AGND CTOUCH	TRACE3
R	P4.2/ TOUTB	P0.10/ TOUTC	P0.12/ SPIH_SCLK	AVDD_RX/ AVDD_TX	RCAL 2	AFE 1	AFE 3	AFE 5	AFE 7	VREF	TIA_O	REF_ EXCITE	AN_C	AN_D	TRST

11714-176

Figure 176. Bump Location (Top View Looking Through Device; Bumps Are Not to Scale)

Table 639. Pin Function Descriptions

Pin No.	Mnemonic	I/O[1]	I/O Supply[2]	GPIO Pull-Up/Down[2]	Description
Power and Ground					
P4	VCCM_ANA	S	VCCM_ANA	N/A	Battery Connection and Analog Circuit Power. Connect VCCM_ANA to the CR2032 battery. VCCM_ANA powers the analog circuits. This pin is connected to VCCM_DIG internally.
H6	VCCM_DIG	S	VCCM_DIG	N/A	Battery Connection and Digital Circuit Power. Connect VCCM_DIG to the CR2032 battery. VCCM_DIG powers the digital circuits. This pin is connected to VCCM_ANA internally.
G2	VBUS	S	VBUS	N/A	5 V USB Supply Voltage.
M1	VBACK	S	VBACK	N/A	RTC Supply. Connect VBACK to the super capacitor.
H15	VDD_IO	S	VDD_IO	N/A	VDD_IO Supply.
E1	VUSB	A	VUSB	N/A	Regulated USB 3.6 V Supply.
R10	VREF	A	N/A	N/A	1.8 V Reference Voltage Decoupling Capacitor Pin.
P10	VBIAS	A	N/A	N/A	1.1 V Bias Voltage Decoupling Capacitor Pin.
K1	DVDD	A	N/A	N/A	1.8 V Digital Regulator Decoupling Capacitor Pin.
R4	AVDD_RX/AVDD_TX	A	AVDD TX/RX	N/A	1.8 V Analog Regulator Decoupling Capacitor Pin for Receiver (Rx)/Transmitter (Tx) Circuits.
K9	AGND_RX/AGND_TX	G	N/A	N/A	Rx/Tx Analog Ground.
K10	AGND_REF	G	N/A	N/A	Reference Ground.
H14	DGND	G	N/A	N/A	Digital Ground.

Pin No.	Mnemonic	I/O[1]	I/O Supply[2]	GPIO Pull-Up/Down[2]	Description
G6	DGND1	G	N/A	N/A	Digital Ground.
F7	DGND2/LCD_GND	G	N/A	N/A	Digital Ground/Ground for LCD.
F6	DGND USB	G	N/A	N/A	USB Ground. Connect DGND USB to the digital ground plane.
AFE Pins					
P12	AN_A	A	VCCM_ANA	N/A	ADC Mux Input.
P13	AN_B	A	VCCM_ANA	N/A	ADC Mux Input.
R13	AN_C	A	VCCM_ANA	N/A	ADC Mux Input.
R14	AN_D	A	VCCM_ANA	N/A	ADC Mux Input.
P5	RCAL 1	A	VCCM_ANA	N/A	Terminal A of Calibration Resistor. Connect RCAL 1 to the switch matrix.
R5	RCAL 2	A	VCCM_ANA	N/A	Terminal B of Calibration Resistor. Connect RCAL 2 to the switch matrix.
R6	AFE 1	A	VCCM_ANA	N/A	Uncommitted AFE Pin 1.
P6	AFE 2	A	VCCM_ANA	N/A	Uncommitted AFE Pin 2.
R7	AFE 3	A	VCCM_ANA	N/A	Uncommitted AFE Pin 3.
P7	AFE 4	A	VCCM_ANA	N/A	Uncommitted AFE Pin 4.
R8	AFE 5	A	VCCM_ANA	N/A	Uncommitted AFE Pin 5.
P8	AFE 6	A	VCCM_ANA	N/A	Uncommitted AFE Pin 6.
R9	AFE 7	A	VCCM_ANA	N/A	Uncommitted AFE Pin 7.
P9	AFE 8	A	VCCM_ANA	N/A	Uncommitted AFE Pin 8.
P11	TIA_I	A	VCCM_ANA	N/A	Transimpedance Amplifier Input. Connect the IV resistor to this pin.
R11	TIA_O	A	VCCM_ANA	N/A	Transimpedance Amplifier Output. Connect the IV resistor to this pin.
R12	REF_EXCITE	A	VCCM_ANA	N/A	Gated Precision Reference Voltage.
Debug Interface					
J1	TDO-SWO/P0.6/UTX	I/O	VCCM_DIG	Pull-up	JTAG Serial Data Output or Serial Wire Trace Output/GPIO/UART_TX. This is a multifunction pin.
J2	TDI/P0.7/URX	I/O	VCCM_DIG	Pull-up	JTAG Serial Data Input/GPIO/UART_RX. This is a multifunction pin.
H1	TMS-SWDIO/P0.8	I/O	VCCM_DIG	Pull-up	JTAG Test Mode Select or Serial Wire Data/GPIO. This is a multifunction pin.
H2	TCK-SWCLK/P0.9	I/O	VCCM_DIG	Pull-down	JTAG Test Clock or Serial Wire Clock/GPIO. This is a multifunction pin.
R15	TRST	I	VCCM	Pull-up	Trace Reset.
M15	TRACECLK	O	VCCM	N/A	Trace Clock.
M14	TRACE0	O	VCCM	N/A	Trace Data 0.
N15	TRACE1	O	VCCM	N/A	Trace Data 1.
N14	TRACE2	O	VCCM	N/A	Trace Data 2.
P15	TRACE3	O	VCCM	N/A	Trace Data 3.
SPI H					
R3	P0.12/SPIH_SCLK	I/O	VCCM_DIG	Pull-up	GPIO/Serial Port H Clock. This is a dual function pin.
P2	P0.13/SPIH_MISO	I/O	VCCM_DIG	Pull-up	GPIO/Serial Port H MISO. This is a dual function pin.
P3	P0.14/SPIH_MOSI	I/O	VCCM_DIG	Pull-up	GPIO/Serial Port H MOSI. This is a dual function pin.
M2	P0.15/SPIH_CS	I/O	VCCM_DIG	Pull-up	GPIO/Serial Port H Chip Select (Active Low). This is a dual function pin.
Other Serial Ports					
F14	P3.0/SPI0_SCLK	I/O	VDD_IO	Pull-up	GPIO/SPI 0 SCLK. This is a dual function pin.
G14	P3.1/SPI0_MISO	I/O	VDD_IO	Pull-up	GPIO/SPI 0 MISO. This is a dual function pin.
F15	P3.2/SPI0_MOSI	I/O	VDD_IO	Pull-up	GPIO/SPI 0 MOSI. This is a dual function pin.
F10	P3.3/SPI0_CS	I/O	VDD_IO	Pull-up	GPIO/SPI 0 Chip Select (Active Low). This is a dual function pin.

Pin No.	Mnemonic	I/O[1]	I/O Supply[2]	GPIO Pull-Up/Down[2]	Description
G10	P3.4/I2CSCL/SPI1_SCLK	I/O	VDD_IO	Pull-up	GPIO (External Interrupt 7)/I²C Clock/SPI 1 SCLK. This is a multifunction pin.
H10	P3.5/I2CSD/SPI1_MISO	I/O	VDD_IO	Pull-up	GPIO/I²C Data/SPI 1 MISO. This is a multifunction pin.
G15	P3.6/UTX/TOUTB/SPI1_MOSI	I/O	VDD_IO	Pull-up	GPIO/UART Tx/Timer B Output/SPI 1 MOSI. This is a multifunction pin.
J10	P3.7/URX/TOUTC/SPI1_CS	I/O	VDD_IO	Pull-up	GPIO/UART Rx/Timer C Output/SPI 1 Chip Select (Active Low). This is a multifunction pin.
USB					
F1	USB DM	I/O	VCCM_DIG	N/A	USB Data −.
G1	USB DP	I/O	VCCM_DIG	N/A	USB Data +.
CapTouch Interface					
K15	P0.0/CAPT_A	A	VCCM_DIG	Pull-up	GPIO (External Interrupt 1)/CapTouch A. This is a dual function pin.
J15	P0.1/CAPT_B	A	VCCM_DIG	Pull-up	GPIO (External Interrupt 2)/CapTouch B. This is a dual function pin.
L15	P0.2/CAPT_C	A	VCCM_DIG	Pull-up	GPIO (External Interrupt 3)/CapTouch C. This is a dual function pin.
K14	P0.3/CAPT_D	A	VCCM_DIG	Pull-up	GPIO (External Interrupt 4)/CapTouch D. This is a dual function pin.
J14	P0.4/CAPT_E	A	VCCM_DIG	Pull-up	GPIO (External Interrupt 5)/CapTouch E. This is a dual function pin.
L14	P0.5/CAPT_F	A	VCCM_DIG	Pull-up	GPIO (External Interrupt 6)/CapTouch F. This is a dual function pin.
P14	AGND CTOUCH	G	N/A	N/A	Capacitance to Digital Converter AC Shield.
System Clocks					
P1	LF_XTAL1	A	RTC_VBACK	N/A	32 kHz XTAL Pin.
N1	LF_XTAL2	A	RTC_VBACK	N/A	32 kHz XTAL Pin.
D1	HF_XTAL1	A	DVDD	N/A	16 MHz XTAL Pin.
C1	HF_XTAL2	A	DVDD	N/A	16 MHz XTAL Pin.
Display					
E2	VLCD FLY1	A	VLCD VDD	N/A	LCD Flying Capacitor Top Plate.
F2	VLCD FLY2	A	VLCD VDD	N/A	LCD Flying Capacitor Bottom Plate.
D2	VLCDVDD	S	N/A	N/A	Full-Scale LCD Voltage Output or VLCD Supply.
C2	V_LCD_13	A	VLCD VDD	N/A	One-Third (1/3) LCD Voltage. Leave this pin as no connect.
B1	V_LCD_23	A	VLCD VDD	N/A	Two-Thirds (2/3) LCD Voltage. Leave this pin as no connect.
B2	P2.0/COM0	I/O	VLCD VDD	Pull-up	GPIO/Common Output 0 for LCD Back Plane (COM 0). This is a dual function pin.
A2	P2.1/COM1/RESX	I/O	VLCD VDD	Pull-up	GPIO/COM 1/Parallel Display Interface (PDI) Reset. This is a multifunction pin.
B3	P2.2/COM2/CSX	I/O	VLCD VDD	Pull-up	GPIO/COM 2/PDI Chip Select. This is a multifunction pin.
A3	P2.3/COM3/DCX	I/O	VLCD VDD	Pull-up	GPIO/COM 3/PDI Data Select. This is a multifunction pin.
B4	P2.4/S1/RWX-RDX	I/O	VLCD VDD	Pull-up	GPIO/Segment Driver 1 (SEG 1)/PDI R/WX or RDX. This is a multifunction pin.
A4	P2.5/S2/ECLOCK-WRX	I/O	VLCD VDD	Pull-up	GPIO/SEG 2/PDI E Clock Output (Motorola Bus Mode) or PD Write Select (Intel® Bus Mode). This is a multifunction pin.
A5	P1.0/S3/D0/SCL	I/O	VLCD VDD	Pull-down	GPIO/SEG 3/PDI D0/PDI Serial Port Clock. This is a multifunction pin.
B5	P1.1/S4/D1/DOUT	I/O	VLCD VDD	Pull-down	GPIO/SEG 4/PDI D1/PDI Serial Port Data Output. This is a multifunction pin.

Pin No.	Mnemonic	I/O[1]	I/O Supply[2]	GPIO Pull-Up/Down[2]	Description
A6	P1.2/S5/D2/DIN	I/O	VLCD VDD	Pull-down	GPIO/SEG 5/PDI D2/PDI Serial Port Data Input. This is a multifunction pin.
B6	P1.3/S6/D3	I/O	VLCD VDD	Pull-down	GPIO/SEG 6/PDI D3. This is a multifunction pin.
A7	P1.4/S7/D4	I/O	VLCD VDD	Pull-down	GPIO/SEG 7/PDI D4. This is a multifunction pin.
B7	P1.5/S8/D5	I/O	VLCD VDD	Pull-down	GPIO/SEG 8/PDI D5. This is a multifunction pin.
A8	P1.6/S9/D6	I/O	VLCD VDD	Pull-down	GPIO/SEG 9/PDI D6. This is a multifunction pin.
B8	P1.7/S10/D7	I/O	VLCD VDD	Pull-down	GPIO/SEG 10/PDI D7/System Clock Output. This is a multifunction pin.
A9	P1.8/S11/D8	I/O	VLCD VDD	Pull-down	GPIO/SEG 11/PDI D8. This is a multifunction pin.
B9	P1.9/S12/D9	I/O	VLCD VDD	Pull-down	GPIO/SEG 12/ PDI D9. This is a multifunction pin.
A10	P1.10/S13/D10	I/O	VLCD VDD	Pull-down	GPIO/SEG 13/PDI D10. This is a multifunction pin.
B10	P1.11/S14/D11	I/O	VLCD VDD	Pull-down	GPIO/SEG 14/PDI D11. This is a multifunction pin.
A11	P1.12/S15/D12	I/O	VLCD VDD	Pull-down	GPIO/SEG 15/PDI D12. This is a multifunction pin.
B11	P1.13/S16/D13	I/O	VLCD VDD	Pull-down	GPIO/SEG 16/PDI D13. This is a multifunction pin.
A12	P1.14/S17/D14	I/O	VLCD VDD	Pull-down	GPIO/SEG 17/PDI D14. This is a multifunction pin.
B12	P1.15/S18/D15	I/O	VLCD VDD	Pull-down	GPIO/SEG 18/PDI D15. This is a multifunction pin.
D15	P2.6/S19/TE	I/O	VLCD VDD	Pull-down	GPIO/SEG 19/TE. This is a multifunction pin.
C15	P2.7/S20/TOUTA	I/O	VLCD VDD	Pull-down	GPIO/SEG 20/Timer A Output. This is a multifunction pin.
B15	P2.8/S21	I/O	VLCD VDD	Pull-down	GPIO/SEG 21. This is a dual function pin.
A14	P2.9/S22	I/O	VLCD VDD	Pull-down	GPIO/SEG 22. This is a dual function pin.
A13	P2.10/S23	I/O	VLCD VDD	Pull-down	GPIO/SEG 23. This is a dual function pin.
B13	P2.11/S24	I/O	VLCD VDD	Pull-down	GPIO/SEG 24. This is a dual function pin.
B14	P2.12/S25	I/O	VLCD VDD	Pull-up	GPIO/SEG 25. This is a dual function pin.
D14	P2.13/S26	I/O	VLCD VDD	Pull-up	GPIO/SEG 26. This is a dual function pin.
E15	P2.14/S27	I/O	VLCD VDD	Pull-up	GPIO/SEG 27. This is a dual function pin.
A15	P2.15/S28	I/O	VLCD VDD	Pull-up	GPIO/SEG 28. This is a dual function pin.
F8	P3.8/S29	I/O	VLCD VDD	Pull-up	GPIO/SEG 29. This is a dual function pin.
F9	P3.9/S30	I/O	VLCD VDD	Pull-up	GPIO/SEG 30. This is a dual function pin.
C14	P3.10/S31	I/O	VLCD VDD	Pull-up	GPIO/SEG 31. This is a dual function pin.
E14	P3.11/S32	I/O	VLCD VDD	Pull-up	GPIO/SEG 32. This is a dual function pin.
Miscellaneous Digital Input/Output					
K8	RESETX	I	VCCM_DIG	Pull-up	Reset Pin (Active Low).
L1	P4.0/I2CSCL	I/O	VCCM_DIG	Pull-up	GPIO (External Interrupt 0)/I²C Clock. This is a dual function pin.
L2	P4.1/I2CSD	I/O	VCCM_DIG	Pull-up	GPIO/I²C Data. This is a dual function pin.
R1	P4.2/TOUTB	I/O	VCCM_DIG	Pull-up	GPIO/Timer B Output. This is a dual function pin.
R2	P0.10/TOUTC	I/O	VCCM_DIG	Pull-up	GPIO (External Interrupt 8)/Timer C Output. This is a dual function pin.
K2	P0.11	I/O	VCCM_DIG	Pull-up	GPIO (External Clock Input Pin).
N2	BOOT	I	VCCM_DIG	Pull-down	The device enters serial download mode if this pin is held high during, and for a short time after, a reset. It executes user code after any reset event or if the pin is low.
A1	DNC		N/A	N/A	Do Not Connect. Leave this pin floating.
Audio					
K6	P3.12/BEEP/BMCLK	I/O	VCCM_DIG	Pull-down	GPIO/Beeper Output Positive/I²S Bit Clock. This is a multifunction pin.
K7	P3.13/BEEPX/SDATA	I/O	VCCM_DIG	Pull-down	GPIO/Beeper Output Negative/I²S Serial Data Output. This is a multifunction pin.
J6	P3.14/LRCLK	I/O	VCCM_DIG	Pull-down	GPIO/I²S Frame Clock. This is a dual function pin.

[1] S is supply, A is analog input, I is digital input, O is digital output, I/O is digital input/output, and G is ground.
[2] N/A means not applicable.

DIGITAL INPUT/OUTPUT PADS

All digital input/output pads must have programmable weak pull-up or weak pull-down resistors. The input/output voltage for the digital pads is VCCM, except for the USB pins whose input/output voltage is VUSB, in which case it is the 3.3 V regulated USB supply.

DISCRETE CAPACITOR REQUIREMENTS

The following discrete capacitors are required.

Table 640. Discrete Capacitor Requirements

Location	Capacitor Value	Domain	Description
VLCDVDD to DGND	470 nF	Digital	Standard decoupling, 0402 Example: GRM155R60J474KE19D
VBUS to DGND	4.7 µF	Digital	LDO input, 0603
VUSB to DGND	220 nF	Digital	Standard decoupling, 0402
VCCM_DIG to DGND	4.7 µF	Digital	Standard decoupling, 0603
DVDD to DGND	470 nF	Digital	Standard decoupling, 0402
VDD_IO to DGND	470 nF	Digital	Standard decoupling, 0402
VBACK		Digital	Supercapacitor pin
LF_XTAL1 to DGND	15 pF	Digital	Capacitor, 0603, GRM0335C1E150JD01D
LF_XTAL2 to DGND	15 pF	Digital	Capacitor, 0603, GRM0335C1E150JD01D
HF_XTAL1 to DGND		Digital	Capacitors integrated on crystal
HF_XTAL1 to DGND		Digital	Capacitors integrated on crystal
VCCM ANA to AGND	10 µF	Analog	Standard decoupling, 0603
AVDD_TX/RX to AGND	0.47 µF	Analog	Standard decoupling, 0402
VREF to AGND	4.7 µF	Analog	GRM188R60J475KE19, Murata, 4.7 µF, X5R, 6.3 V, 0603
VBIAS to AGND	0.47 µF	Analog	Standard decoupling, 0402

RESETX

The device can be reset with an external pin, RESETX. The RESETX pin is an active low reset. The pin can be left unconnected in the application, and it has a weak pull-up resistor at the input enabled by default.

ESD REQUIREMENTS

The device has an HBM specification of ±2 kV and an ESD machine model of ±200 V on all pins. This is a requirement. For IEC60601 ESD requirements, it is advised that external ESD protection devices be used on the AFE sensor connected pins.

REFERENCES

ACRONYMS AND ABBREVIATIONS

Table 641.

Acronym/Abbreviation	Description
ACM	Abstract class module
ADC	Analog-to-digital converter
AFE	Analog front end
AHB	AMBA high performance bus
AMBA	Advanced microcontroller bus architecture
APB	Advanced performance bus
ARM	Advanced RISC machine
C2V	Capacitance to voltage
CDC	Communications device class
CPU	Central processing unit
CRC	Cyclic redundancy check
DAC	Digital-to-analog converter
DBus	Data bus
DCC	Duty cycle correction
DFT	Discrete Fourier transform
DMA	Direct memory access
DSP	Digital signal processing
DWT	Data watchpoint and trigger
FIFO	First in, first out
ETM	Embedded trace Macrocell
FPB	Flash patch and breakpoint
GPIO	General-purpose input and output
HID	Human interface device
I^2C	Inter IC
I^2S	Inter IC sound bus
ITM	Instruction trace module
JTAG	Joint test action group
IIR	Infinite impulse response
IRQ	Interrupt
LCD	Liquid crystal display
LDO	Low drop out
LSB	Least significant byte/bit
MCU	Microcontroller unit
MMR	Memory mapped register
MSB	Most significant byte/bit
MSC	Mass storage class
NVIC	Nested vectored interrupt controller
PCLK	Peripheral clock
PFD	Phase frequency detector
PGA	Programmable gain amplifier
PHY	Physical layer
PLL	Phase locked loop
PMU	Power management unit
POR	Power-on reset
PSM	Power supply monitor
PWM	Pulse width modulation
RAM	Random access memory
RISC	Reduced instruction set computer
RTC	Real-time clock

Acronym/Abbreviation	Description
SNR	Signal-to-noise ratio
SOF	Start of frame
SPI	Serial peripheral interface
SPLL	System PLL
SWD	Serial wire debug
TIA	Transimpedance amplifier
UART	Universal asynchronous receiver-transmitter
UPLL	USB PLL
USB	Universal serial bus
WDT	Watchdog timer
WUT	Wake-Up Timer

RELATED DOCUMENTS

Related documents include the following:

- ARM Cortex-M3 Technical Reference Manual
- ARM Cortex-M3 Devices Generic User Guide
- ARM Cortex-M3 Errata
- USB Specification Revision 2.0, Chapter 9
- I^2C Bus Specification, Version 2.1
- I^2S Standard
- IEEE Standards: (IEEE 802.3)
- MIPI DBI Specification, Version 2.00

LIMITATIONS ON USE AND LIABILITY

This hardware reference manual provides instructions for using the features and functionality of the ADuCM350. When the ADuCM350 is used with the EVAL-ADuCM350EBZ evaluation board for biometric applications, the following cautions apply: in addition to the terms of use contained in the evaluation board user guide(s), it is understood and agreed to that the evaluation board or design must not be used for diagnostic purposes and must not be connected to a human being or animal. This evaluation board is provided for evaluation and development purposes only. It is not intended for use or as part of an end product. Any use of the evaluation board or design in such applications is at your own risk and you shall fully indemnify Analog Devices, Inc., its subsidiaries, employees, directors, officers, servants and agents for all liability and expenses arising from such unauthorized usage. You are solely responsible for compliance with all legal and regulatory requirements connected to such use.

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).