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

ADuCM300 Automotive Qualified, Dual-Channel, Precision ADCs with LIN2.2 Slave Interface

SCOPE

This reference manual provides a detailed description of the functionality and features of the ADuCM300. The information found in this document is relevant for Silicon Revision A60.

For full specifications, see the ADuCM300 data sheet, which must be consulted in conjunction with this reference manual when working with the device.

Note that throughout this hardware reference manual, multifunction pins are referred to either by the entire pin name or by a single function of the pin when only that function is relevant. For more information, see the ADuCM300 data sheet.

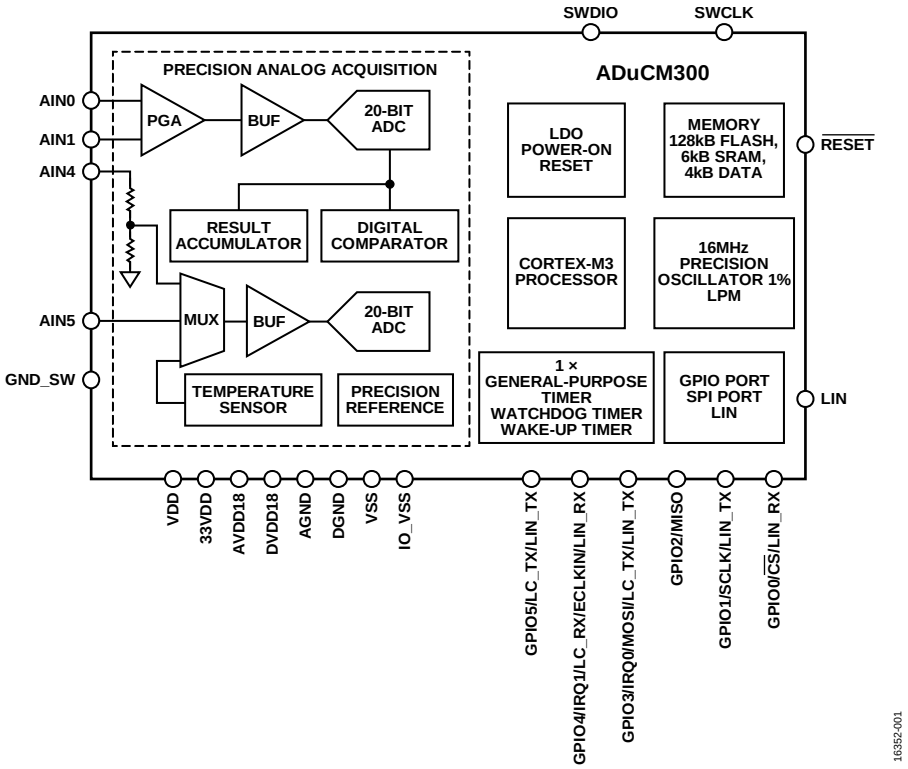

Figure 1. ADuCM300

TABLE OF CONTENTS

REVISION HISTORY

6/2019—Rev. 0 to Rev. A

Changes to Overview Section and Precision Measurement Section....5
Change to Power Section....6
Changes to Power Supply Support Circuits Section....14
Change to Direct LIN Interface (External Transceiver Mode) Section....24
Change to Programmable Gain Amplifier Section....55
Change to LIN Diagnostic Frame Support Section....101

6/2019—Revision 0: Initial Version

USING THE ADUCM300 HARDWARE REFERENCE MANUAL

The tables in this section provide information needed to properly use this reference manual.

NUMBER NOTATIONS

Table 1. Number Notations

Notation	Description
Bit N	Bits are numbered in little endian format, that is, the LSB of a number is referred to as Bit 0.
V[x:y]	Bit field representation for Bit x to Bit y of a value or a field (V).
0xNN	Hexadecimal (Base 16) numbers are preceded by the 0x prefix.
0bNN	Binary (Base 2) numbers are preceded by the 0b prefix.
NN	Decimal (Base 10) numbers are represented using no additional prefixes or suffixes.
0bXX	An X in the binary number means that the contents of the bit are not defined.
0xXX	An X in the hexadecimal number means that the contents of the four bits are not defined.

REGISTER ACCESS CONVENTIONS

Table 2. Register Access Conventions

Mode	Description
Read/Write	Memory location has read and write access.
Read	Memory location is read access only. A read always returns 0, unless otherwise specified.
Write	Memory location is write access only.

ADUCM300 OVERVIEW

The ADuCM300 is a fully integrated, 8 kSPS, data acquisition system that incorporates dual, high performance, multichannel, Σ-Δ analog-to-digital converters (ADCs), a 32-bit Arm® Cortex®-M3 processor, and flash memory. The ADuCM300 has a 128 kB program flash, 4 kB data flash, and 6 kB static random access memory (SRAM).

The ADuCM300 is fully qualified for general automotive sensing applications and can interface to multiple external precision sensors in a system solution that also integrates a programmable Cortex-M3 microcontroller to postprocess the sensor signals on-chip, before transmitting the results off chip via industry-standard wired interfaces.

The ADuCM300 integrates all of the required features to precisely and intelligently interface, monitor, process, and diagnose a precision sensing system over a wide range of operating conditions.

The ADuCM300 operates from an on-chip, 16.384 MHz, high frequency oscillator that supplies the system clock. This clock is routed through a programmable clock divider from which the core clock operating frequency is generated. The device also contains a 32.768 kHz, low frequency oscillator for low power operation.

On-chip, low dropout (LDO) regulators generate the supply voltage for two integrated Σ-Δ ADCs and the microcontroller subsystem. The analog subsystem consists of an ADC with a programmable gain amplifier (PGA) to allow the monitoring of various input voltage ranges. The subsystem also includes a precision, on-chip reference.

The ADuCM300 integrates a range of on-chip peripherals that can be configured under core software control as required in the application. These peripherals include a serial peripheral interface (SPI) input and output communication controller, six general-purpose input and output (GPIO) pins, one general-purpose timer, a wake-up timer (WUT), and a watchdog timer (WDT).

The ADuCM300 can be used in in applications where low power operation is critical. The microcontroller core can be configured in a normal operating mode, resulting in an overall system current consumption of <18.5 mA when all peripherals are active. The device can also be configured in a number of low power operating modes under direct program control, consuming <100 µA. The ADuCM300 includes a local interconnect network (LIN) physical interface for single wire, high voltage communications in automotive environments. Refer to the ADuCM300 data sheet.

MAIN FEATURES OF THE ADUCM300

The following sections provide brief descriptions of the features of the ADuCM300. Refer to the ADuCM300 data sheet for full details.

Precision Measurement

The device has the following precision measurement features:

- High precision ADCs
- Dual-channel, simultaneous sampling, 20-bit, Σ-Δ ADCs (minimizing gain switching), ADC0 and ADC1
- A programmable ADC conversion rate from 4 Hz to 8 kHz
- An on-chip, ±5 ppm/°C voltage reference, typical
- ADC0, which has a fully differential, buffered input, programmable gain, with an absolute input voltage range of −200 mV to +300 mV and a digital comparator with a result accumulator
- ADC1, which has external and on-chip temperature sensor options, and a buffered, on-chip attenuator.

Microcontroller

The device has the following microcontroller related features:

- Cortex-M3, 32-bit processor
- Serial wire download and debug
- 16.384 MHz high frequency oscillator
- 32.768 kHz low frequency oscillator for low power operation

Memory

The device has the following memory related features:

- 128 kB program Flash/EE memory options
- 4 kB data Flash/EE memory
- 6 kB static random access memory (SRAM)
- 10,000 cycle Flash/EE endurance
- 20 year Flash/EE retention
- In circuit download via serial wire and LIN
- Error correction code (ECC) available on all flash and SRAM memories

Power

See the ADuCM300 data sheet for exact power consumption.

On-Chip Peripherals

The device incorporates seven on-chip peripherals. The on-chip peripherals are on-chip power-on reset, one general-purpose timer, a wake-up timer, a watchdog timer, a GPIO port, an SPI input/output, and a LIN transceiver that is compliant to LIN 2.2 and Society of Automotive Engineers (SAE) J2602-2.

Package

The device is packaged in a 32-lead, 6 mm × 6 mm LFCSP.

ADDITIONAL DOCUMENTS REQUIRED

ADuCM300 related reference documentation include the following:

- ADuCM300 data sheet
- LIN2.2A specification, December 2010
- AN-946 Application Note
- AN-609 Application Note

CORTEX-M3 PROCESSOR

The following sections provide brief descriptions of the features of the Cortex-M3 processor.

CORTEX-M3 PROCESSOR FEATURES

High Performance

The Cortex-M3 processor high performance features include the following:

- Operates up to 1.25 Dhrystone million instructions per second (DMIPS)/MHz.
- Many instructions, including integer 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

The Cortex-M3 processor has low standby current and a core implemented using advanced clock gating. Therefore, only the actively used logic consumes dynamic power. The Cortex-M3 processor also has two sleep modes (sleep mode and deep sleep mode) that provide different levels of power saving. The SLEEPDEEP bit of the system control register selects the sleep mode.

Advanced Interrupt Handling

The nested vectored interrupt controller (NVIC) greatly reduces interrupt latency because additional software is not needed to determine which interrupt handler to serve. In addition, additional software is not needed to set up nested interrupt support.

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

The Cortex-M3 processor also has dynamic priority controls for each interrupt, latency reduction using late arrival interrupt acceptance and tail chain interrupt entry, and immediate execution of a nonmaskable interrupt request.

System Features

The Cortex-M3 processor system has support for bit band operation, unaligned data access, and advanced fault handling features that include various exception types and fault status registers.

Debug Support

The Cortex-M3 processor has debug support features that include serial wire debug port (SWDP) interfaces, a flash patch and breakpoint (FPB) unit for implementing breakpoints that is limited to two active breakpoints, and a data watchpoint and trigger (DWT) unit for implementing watchpoints, trigger resources, and system profiling. The Cortex-M3 is limited to one active watchpoint.

CORTEX-M3 PROCESSOR OVERVIEW

The ADuCM300 contains an embedded Cortex-M3 processor, Revision r2p0 (AT420). The Cortex-M3 provides a high performance, low cost platform that meets the system requirements of minimal memory implementation, reduced pin count, and low power consumption, while delivering high computational performance and fast system response to interrupts.

CORTEX-M3 PROCESSOR OPERATION

The following sections describe the implementation and configuration of the Cortex-M3 component in the ADuCM300.

Serial Wire Debug (SWD)

The ADuCM300 supports the serial wire interface via the SWCLK pin and the SWDIO pin. This device does not support the 5-wire joint test action group (JTAG) interface.

ROM Table

The ADuCM300 implements the default read only memory (ROM) table.

NVIC Interrupts

The Cortex-M3 processor includes an interrupt controller, the NVIC. The NVIC provides nested interrupt support, vectored interrupt support, dynamic priority changes support, and interrupt masking.

In addition, the NVIC also has a nonmaskable interrupt (NMI) input. The NVIC is implemented on the ADuCM300 and is described in more detail in the Nested Vectored Interrupt Controller section.

Wake-Up Interrupt Controller (WIC)

The WIC provides a device configuration with the lowest possible power-down current.

More information is available in the Power Management Unit section.

If the device enters a power saving mode when servicing an interrupt, the device can only then wake up from a higher priority interrupt source.

RELATED DOCUMENTATION

Additional Cortex-M3 related documentations are

- Cortex-M3 Devices, *Generic User Guide*, Arm DUI 0552A (ID121610)
- Cortex-M3 Revision r2p0 *Technical Reference Manual* (DDI 0337)
- Arm Processor Cortex-M3 (AT420) and Cortex-M3 with ETM (AT425) Software Developers Errata Notice
- Armv7-M Architecture Reference Manual (DDI 0403)
- Arm Debug Interface Architecture Specification v5 (IHI 0031)

EXCEPTIONS AND INTERRUPTS

CORTEX-M3 AND FAULT MANAGEMENT

The Cortex-M3 supports a number of system exceptions and peripherals and external interrupts, as listed in Table 3 and Table 4.

Table 3. System Exceptions

Vector Table Position	Type	Priority	Description
1	Reset	−3 (highest)	Any reset.
2	Reserved	Not applicable	
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 and memory related faults. This exception occurs if there is a problem reading or writing with the advanced high performance bus (AHB). For example, uncorrectable ECC errors cause this exception to occur.
6	Usage fault	Programmable	Faults such as undefined instruction executed or illegal state transition attempt.
7 to 10	Reserved	Not applicable	
11	SVCALL	Programmable	System service call with supervisor calls instruction.
12	Debug monitor	Programmable	Debug monitor (breakpoint, watchpoint, or external debug requests).
13	Reserved	Not applicable	
14	PENDSV	Programmable	Pendable request for system service.
15	SYSTICK	Programmable	System tick timer.
16	Interrupt request (IRQ)	Programmable	ADuCM300 interrupts controlled by the NVIC.

NESTED VECTORED INTERRUPT CONTROLLER

The ADuCM300 interrupts are controlled by the NVIC, and four levels of priority are available. NVIC interrupts can be enabled and disabled by writing to their corresponding interrupt set enable or interrupt clear enable register bit field. Only a limited number of interrupts can wake up the core from low power hibernate mode. These interrupts are described in Table 4. When the ADuCM300 wakes up from any mode, the device returns to active mode.

Table 4. Interrupt Vectors

Position No.	Priority	Vector	Wake Up Core from Hibernate Mode
0	Programmable	Wake-up timer interrupt	Yes
1	Programmable	External Interrupt 0 (P0.3)	Yes
2	Programmable	External Interrupt 1 (P0.4)	Yes
3	Programmable	Watchdog timer interrupt	Yes
4	Programmable	General-Purpose Timer 0 interrupt	No
5	Programmable	ADC interrupt	No
6	Programmable	Flash interrupt	No
7	Programmable	SPI interrupt	No
8	Programmable	LIN0: LIN header or frame interrupt	No
9	Programmable	LIN1: LIN error detected interrupt	No
10	Programmable	LIN2: LIN sleep or wake-up event interrupt	Yes
11	Programmable	High voltage interface interrupt	No
12	Programmable	Low frequency oscillator calibration interrupt	No
13	Programmable	SRAM ECC interrupt	No

For the ADuCM300, each interrupt can have eight levels of priority. The priority levels are 0 to 7, where 0 is the highest priority and 7 is the lowest priority. Internally, the highest user programmable priority (0) is treated as fourth priority, after a reset, an NMI, and a hardware fault. Zero is the default priority for all the programmable priorities.

If the same priority level is assigned to two or more interrupts, their hardware priority (the lower the position number, the higher the priority, as shown in Table 4) determines the order in which the processor activates them. For example, if both the ADC and SPI are Priority Level 1, the ADC has higher priority.

SETTING INTERRUPT PRIORITIES

The Cortex-M3 IPR0 register to IPR3 register control the interrupt priority settings for the ADuCM300. The user can adjust these registers to change the default interrupt priority and to create a user specific interrupt vector table to suit the application of the user.

Every interrupt has eight possible priority levels, with 0 being the highest priority and 7 being the lowest priority setting. Each interrupt priority register supports four interrupt sources. Table 5 uses IPR0 as an example to explain the relevant bits.

The IPR1 to IPR3 interrupt priority registers are configured similarly, with default priority according to Table 4.

Table 5. Bit Descriptions for the Interrupt Priority Registers, IPR0

Bits	Name	Description
[31:29]	Watchdog timer	000: highest interrupt priority level for watchdog timer. 111: lowest interrupt priority level. The watchdog timer must be configured to cause a reset.
[28:24]	Reserved	Reserved. These bits must be set to 0.
[23:21]	EXTINT1	000: highest interrupt priority level for external IRQ1. 111: lowest interrupt priority level.
[20:16]	Reserved	Reserved. These bits must be set to 0.
[15:13]	EXTINT0	000: highest interrupt priority level for external IRQ0. 111: lowest interrupt priority level.
[12:8]	Reserved	Reserved. These bits must be set to 0.
[7:5]	Wake-up timer	000: highest interrupt priority level for wake-up timer. 111: lowest interrupt priority level.
[4:0]	Reserved	Reserved. These bits must be set to 0.

CORTEX-M3 AND NVIC REGISTERS

The interrupt set enable register (ISER0) and the interrupt clear enable register (ICER0) enable and disable the interrupts. The interrupt set pending register (ISPR0) and the interrupt clear pending register (ICPR0) pend the interrupts. The ISER0 register, ICER0 register, ISPR0 register, and ICPR0 register use a write 1 to enable and a write 1 to clear policy. Each bit in the interrupt set enable register corresponds to one of 32 interrupts. Setting a bit in the interrupt set enable register enables the interrupt.

When the enable bit of a pending interrupt is set in the interrupt set pending register, the processor activates the interrupt based on the priority of the interrupt. However, if the corresponding bit is clear in the set enable register, asserting the interrupt signal pends the interrupt, but it is not possible to activate the interrupt, regardless of its priority.

Pending the interrupt means that, if the interrupt occurs, it is possible for the user code to investigate the interrupt set pending register and verify whether the interrupt occurs. Therefore, a disabled interrupt can serve as a latched general-purpose bit.

Reading and clearing interrupts occur without invoking an interrupt.

Clear the enable state by writing a 1 to the corresponding bit in the interrupt clear enable register. This write also clears the corresponding bit in the interrupt set enable register. Writing to the interrupt clear pending register has no effect on an interrupt that is active, unless the interrupt is also pending. If an interrupt is active when it is disabled, the interrupt remains in its active state until cleared by a reset or an exception return.

The NVIC is an integral part of the Cortex-M3 microprocessor. The NVIC is notified that an interrupt is generated from peripherals within one clock cycle of the clock used by this peripheral, unless otherwise noted.

Table 6. Cortex-M3 Registers and NVIC Registers

Address	Analog Devices Header File Name	Description	Access
0xE000E004	ICTR	Shows the number of interrupt lines that the NVIC supports	Read
0xE000E010	STCSR	SYSTICK control and status register	Read/write
0xE000E014	STRVR	SYSTICK reload value register	Read/write
0xE000E018	STCVR	SYSTICK current value register	Read/write
0xE000E01C	STCR	SYSTICK calibration value register	Read
0xE000E100	ISER0	Set IRQ0 to IRQ13 enable	Read/write
0xE000E180	ICER0	Clear IRQ0 to IRQ13 enable	Read/write
0xE000E200	ISPR0	Set IRQ0 to IRQ13 pending	Read/write
0xE000E280	ICPR0	Clear IRQ0 to IRQ13 pending	Read/write
0xE000E300	IABR0	IRQ0 to IRQ13 active bits	Read/write
0xE000E400	IPR0	IRQ0 to IRQ3 priority	Read/write
0xE000E404	IPR1	IRQ4 to IRQ7 priority	Read/write
0xE000E408	IPR2	IRQ8 to IRQ11 priority	Read/write
0xE000E40C	IPR3	IRQ12 to IRQ13 priority	Read/write
0xE000ED00	CPUID	Central processing unit identification (CPUID) base register	Read
0xE000ED04	ICSR	Interrupt control and status register	Read/write
0xE000ED08	VTOR	Vector table offset register	Read/write
0xE000ED0C	AIRCR	Application interrupt and reset control register	Read/write
0xE000ED10	SCR	System control register	Read/write
0xE000ED14	CCR	Configuration control register	Read/write
0xE000ED18	SHPR1	System Handlers Register 1	Read/write
0xE000ED1C	SHPR2	System Handlers Register 2	Read/write
0xE000ED20	SHPR3	System Handlers Register 3	Read/write
0xE000ED24	SHCSR	System handler control and state	Read/write
0xE000ED28	CFSR	Configurable fault status	Read/write
0xE000ED2C	HFSR	Hard fault status	Read/write
0xE000ED34	MMAR	Memory manage address	Read/write
0xE000ED38	BFAR	Bus fault address	Read/write
0xE000EF00	STIR	Software trigger interrupt register	Write

EXTERNAL INTERRUPT CONFIGURATION

Two external interrupts are implemented. The 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 from 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 for one core clock cycle minimum to be detected.

The external interrupt detection unit block allows an external interrupt to wake up the device when in hibernate mode.

EXTERNAL INTERRUPT CONFIGURATION MEMORY MAPPED REGISTERS

The interrupt detection unit consists of memory mapped registers (MMRs) contained in the always on section. The MMRs are based at Address 0x40002400.

Table 7. Interrupt Detection Unit Memory Mapped Registers (Base Address 0x40002400)

Offset	Name	Description	Access	Default
0x0020	EI0CFG	External Interrupt Configuration Register 0	Read/write	0x0000
0x0030	EICLR	External interrupt clear register	Read/write	0x0000

External Interrupt Configuration Register 0

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

Table 8. EI0CFG Register Bit Descriptions

Bits	Name	Description
[15:8]	Reserved	Reserved. These bit must be set to 0
7	IRQ1EN	External Interrupt 1 enable bit 0: external Interrupt 1 disabled 1: external Interrupt 1 enabled
[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)
3	IRQ0EN	External Interrupt 0 enable bit 0: external Interrupt 0 disabled 1: external Interrupt 0 enabled
[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)

External Interrupt Clear Register

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

Ensure that the register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary.

Table 9. EICLR Register Bit Descriptions

Bits	Name	Description
[15:2]	Reserved	Reserved. These bits must be set to 0.
1	IRQ1	External Interrupt 1 clear bit 0: cleared by software 1: clear external Interrupt 1 flag
0	IRQ0	External Interrupt 0 clear bit 0: cleared by software 1: clear external Interrupt 0 flag

POWER MANAGEMENT UNIT

POWER MANAGEMENT UNIT FEATURES

Active mode, system halt mode, and hibernate mode are the three power modes available in the ADuCM300.

To enter system halt mode and hibernate mode, the Cortex-M3 uses sleep mode and deep sleep mode as power saving modes. Sleep mode stops the core clock (FCLK). Deep sleep mode stops the FCLK and, in conjunction with the power management unit (PMU), switches off some circuitry such as the system clock (UCLK), the high frequency oscillator, and the high power LDO.

POWER MANAGEMENT UNIT OVERVIEW

The PMU controls the power modes of the ADuCM300. The Cortex-M3 sleep modes are linked to the PMU modes and are described in this section.

The PMU is always powered on. Three power modes are available: a fully functional active mode and two modes that provide a power reduction benefit with a corresponding reduction in functionality. System halt mode provides significant power reduction with further power savings possible in hibernate mode. Table 10 lists all the power modes available. For active mode and hibernate mode, current values and wake-up times are available in the ADuCM300 data sheet.

Table 10. System Power Mode Summary

Mode	Description
Active	Default.
System halt	Gate both the memory clock (HCLK) and peripheral interface clock (PCLK) when Cortex-M3 is in sleep mode.
Hibernate	Gate power to flash block, high frequency oscillator clock is gated, low frequency oscillator is active.

The Arm wait for interrupt (WFI) instruction places the Cortex-M3 in sleep mode (system halt). Program execution is suspended until a nonmasked interrupt occurs. If deep sleep mode is enabled in the system control register of the Cortex-M3, the Cortex-M3 enters deep sleep mode on execution of a WFI. Otherwise, the Cortex-M3 enters sleep mode.

The PMU and Cortex-M3 power modes are summarized in Table 11.

The SPI peripheral must be disabled before entering low power modes where the PCLK (LIN, SPI, high voltage interface and timers) is disabled. Disabling the peripheral resets the state machine of these peripherals and keepswell their configurations.

The debugger must be disconnected to achieve lower power performance.

The following is a typical code example for achieving low power mode (hibernate mode):

```
SCR       = 0x04;       // Enable deep sleep mode in the core
                        //   (system control register (Address 0xE000ED10))
PWRKEY    = 0x4859;
PWRKEY    = 0xF27B;     // PWRMOD keys
PWRMOD    = 0x5;        // Hibernate Mode
__dsb();                // Wait until all memory accesses complete
__wfi();                // Wait for interrupt
```

POWER MANAGEMENT UNIT LDO SWITCHING

Three LDO regulators are integrated on the ADuCM300.

- The high voltage LDO regulates voltages from 12 V to 3.3 V.
- The high power LDO regulates voltages from 3.3 V to 1.8 V and supplies the device in active mode.
- The low power LDO regulates voltages from 3.3 V to 1.8 V and achieves an extremely low quiescent current. However, the low power LDO can only supply very low load currents and is used in hibernate mode.

Switching between the LDOs is done automatically after the change from active mode to hibernate mode is performed.

POWER MANAGEMENT UNIT POWER MODES OPERATION

Power Mode: Active Mode

The system is fully active with all of the clocks described in Figure 3 available, memories and all user enabled peripherals are clocked, and the Cortex-M3 executes instructions. The Cortex-M3 can manage the internal clocks and can be in a partial clock gated state. Clock gating affects only the internal Cortex-M3 processing core. Automatic clock gating is used on all blocks and is transparent to the user. User code can use a WFI command to place the Cortex-M3 in sleep mode. The WFI command is independent of the power mode settings of the PMU. When the ADuCM300 wakes from any of the low power modes, the device returns to active mode.

Power Mode: System Halt Mode

The system gates the flash and SRAM clock (HCLK) and PCLK at an early stage after the Cortex-M3 enters sleep mode. The Cortex-M3 FCLK is active, the NVIC is operational, and activity on this block wakes up the device.

Power Mode: Hibernate Mode

The system gates power to the digital flash memory. All states are retained during this power gating. In hibernate mode, FCLK, HCLK, and PCLK are clock gated at an early stage after the Cortex-M3 has entered deep sleep mode, resulting in a lower leakage current. Due to the clock gating, more time is taken to wake the device from hibernate mode compared with system halt mode. With the FCLK stopped, only the peripherals listed in Table 4 are able to wake up the Cortex-M3. Bit 2 of the Cortex-M3 system control register (Address 0xE000ED10) must be set to 1 to enter deep sleep mode and cleared to 0 for sleep mode by the user.

Table 11. Power Modes Summary

Clock and Power	Active	System Halt	Hibernate
High Power LDO	On	On	Off
Low Power LDO	Off	Off	On
High frequency oscillator	On	On	Off
Low frequency oscillator	On	On	On
Power gate	On	On	Off
UCLK	On	On	Off
FCLK	On	On	Off
HCLK	On	Off	Off
PCLK	On	Off	Off
ADC interface clock (ACLK)	On	On	Off
SRAM	On	On	On
Cortex-M3	Active	Sleep	Deep sleep

POWER SUPPLY SUPPORT CIRCUITS

The ADuCM300 incorporates three on-chip, LDO regulators. One regulator is driven directly from VDD to generate a 3.3 V internal supply, which is used as the supply for two internal 1.8 V LDO regulators. The two 1.8 V LDO regulators allow normal mode (high power LDO) and low power mode (low power LDO) operation. The HP LDO requires two output capacitors on DVDD18 and AVDD18. Refer to the recommended schematic diagram in the ADuCM300 data sheet for details. The effective series resistance (ESR) of the output capacitor affects the stability of the LDO control loop. A capacitor with a low ESR is recommended to ensure the stability of the regulators.

A power-on reset (POR) function is provided to ensure the safe operation of the processor, as well as continuous monitoring of the input signal voltage level. When the supply voltage on VDD reaches a typical operating voltage threshold, a POR signal keeps the Cortex-M3 processor in a reset state for a defined time (see Figure 2). This delay ensures that the regulated power supply voltage (DVDD33) applied to the Cortex-M3 processor and the associated peripherals is greater than the minimum operational voltage, thereby guaranteeing full functionality.

A POR flag is set in the RSTSTA MMR to indicate that a POR reset event has occurred. At voltages below the POR level voltage, the SRAM may be corrupted. The implemented ECC in the SRAM ensures that a corruption can be detected by the kernel. If the SRAM is corrupted, the kernel initializes the entire SRAM to 0x00 and corrects all ECC bits.

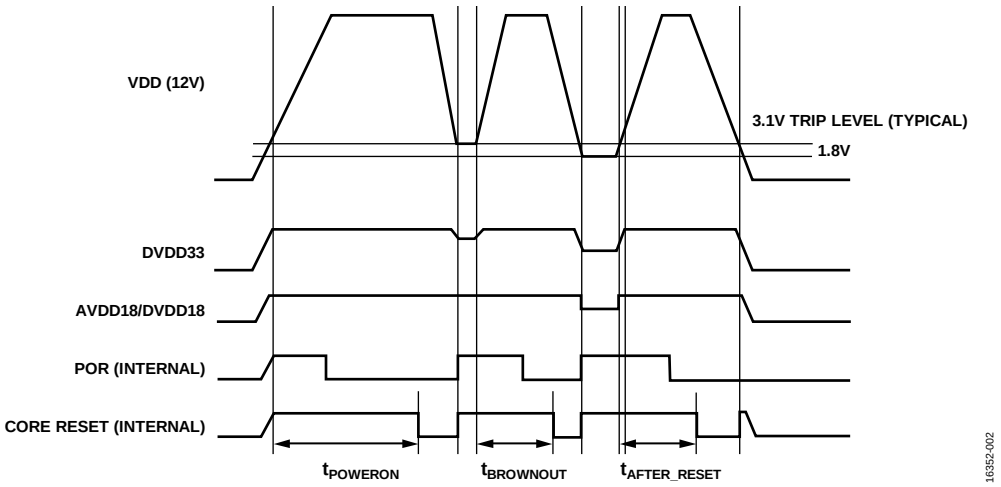

Figure 2. Power Supply Diagram

POWER MODES MEMORY MAPPED REGISTERS

The power modes are controlled by a single register based in the always on section at Address 0x40002400.

Table 12. Power Mode Memory Mapped Register (Base Address 0x40002400)

Offset	Name	Description	Access	Default
0x0000	PWRMOD	Power mode control register	Read/write	0x00
0x0004	PWRKEY	Power mode key register	Read/write	Not applicable

Power Mode Control Register

Address: 0x40002400, Reset: 0x00, Name: PWRMOD

Table 13. PWRMOD Register Bit Descriptions

Bits	Name	Description
7	Reserved	Reserved. This bit must be set to 0.
6	HFOSC_LPM	Enables high or low precision mode in the 16 MHz high frequency oscillator. 0: enable high precision 1%, high precision oscillator mode (HPOSC). 1: enable low precision 3%, low precision oscillator mode (LPOSC).
[5:3]	Reserved	Reserved. These bits must be set to 0.
[2:0]	MOD	Power modes control bits. These bits select the power mode to enter. When read, these bits contain the last power mode value entered by user code. 000: active (normal mode)
		011: system halt. The cortex deep sleep mode has no effect in system halt mode. Before entering system halt mode with ADCs active, deep sleep mode must be enabled in the Cortex-M3 system control register. 101: hibernate. To enter hibernate mode, the user also sets Bit 2 (SLEEPDEEP) in the Cortex-M3 system control register (Address 0xE000ED10) to 1 to place the Cortex-M3 in deep sleep mode. Other: reserved. Only the other three modes must be selected.

Power Mode Key Register

Address: 0x40002404, Reset: not applicable, Name: PWRKEY

Table 14. PWRKEY Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Power modes key register. The PWRMOD register is key protected. Two writes to the key are necessary to change the value in the PWRMOD register: 0x4859, followed by 0xF27B. Following these writes, the PWRMOD register can be written.

SYSTEM CLOCKS

SYSTEM CLOCKS FEATURES

The ADuCM300 integrates two internal clock sources: a high frequency oscillator (HFOSC) and low frequency oscillator (LFOSC).

The high frequency oscillator is a 16.384 MHz oscillator that can operate in two modes, high precision mode and low precision mode with reduced power consumption.

The low frequency oscillator is a 32.768 kHz low power, internal oscillator.

SYSTEM CLOCKS OVERVIEW

The CLKCON MMR controls the clocking source to the ADuCM300. The CLKCON MMR does not control which clocks are enabled.

One of the outputs of the clock generation circuit is directed through the clock divider (CD) circuit (the CD bit in the CLKCON MMR), where the UCLK can be divided down to a minimum of 16.384 MHz ÷ 2 (for high frequency oscillator) by the user. This clock is referred to as the core clock within the ADuCM300, and is the clock that drives the Cortex-M3. This clock is gated even further to all the digital peripherals in the system (PCLK), the NVIC (FCLK), and the ADC (ACLK).

Internally, the system clock is divided into five clocks:

- UCLK system clock
- FCLK for the core
- HCLK for the flash, SRAM
- PCLK for the LIN, SPI, high voltage interface, and timers
- ACLK for the ADC interface

Figure 3 shows all the clocks available and includes clock gates for power management. More information on the clock gates is available in the Power Management Unit section.

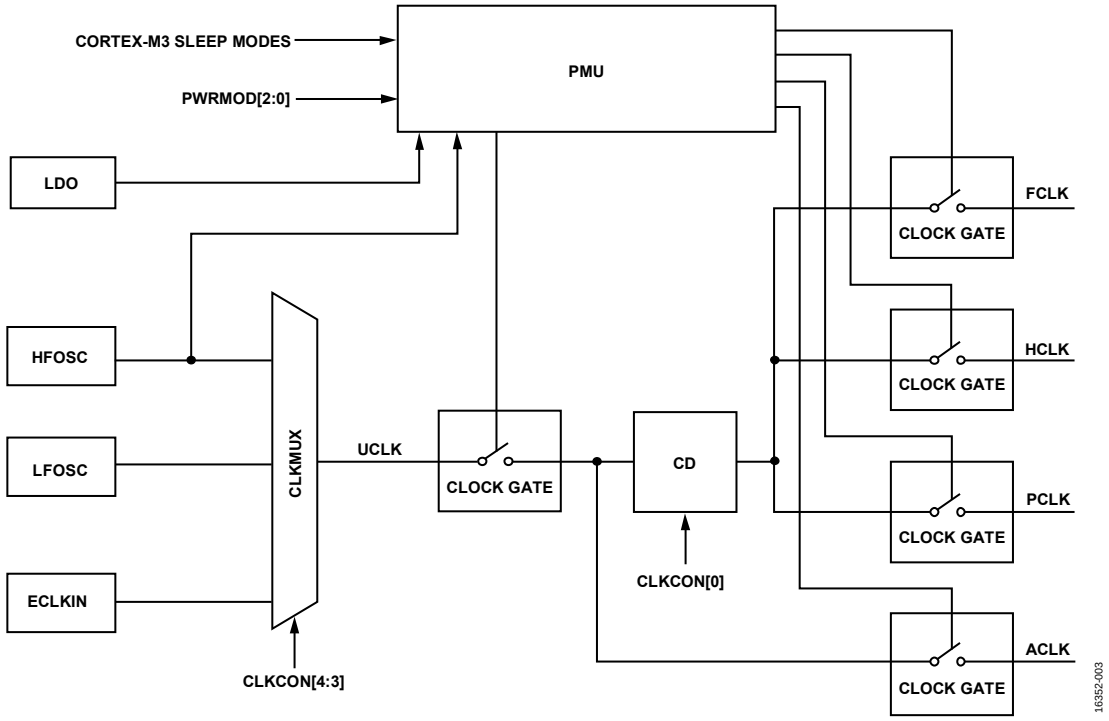

Figure 3. System Clock Architecture Block Diagram

SYSTEM CLOCKS OPERATION

At power-up, the core executes from the high frequency oscillator internal oscillator in high precision mode. User code can select the clock source for the system clock and can divide the clock by a factor of 2. Dividing the clock allows slower code execution and reduced power consumption. The UCLK is also passed to some of the serial peripherals so that the timings are not affected by CD changes.

SYSTEM CLOCKS MEMORY MAPPED REGISTERS

Table 15. Clock Control Memory Mapped Register

Address	Name	Description	Access	Default
0x40002000	CLKCON	System clocks control register	Read/write	0x00E0

System Clocks Control Register

Address: 0x40002000, Reset: 0x00E0, Name: CLKCON

Table 16. CLKCON Register Bit Descriptions

Bits	Name	Description
[15:8]	Reserved	Reserved. These bits must be set to 0.
[7:5]	Reserved	Reserved. These bits must be set to 0b111.
[4:3]	CLKMUX	Clock in multiplexer selection bits. 00: high frequency oscillator (default). 01: high frequency oscillator. 10: low frequency oscillator. 11: external clock (ECLKIN, Pin 8), Pin 4 (P0.4).
[2:1]	Reserved	Reserved. These bits must be set to 0.
0	CD	Clock divide bit. 0: UCLK. 1: UCLK ÷ 2.

HIGH FREQUENCY OSCILLATOR CALIBRATION

The ADuCM300 on-chip, 16 MHz high frequency oscillator can be calibrated using the LIN interface.

A number of MMR registers control the trimming in normal mode and power-down mode (see Table 17). These registers are protected and are writable only following a write to the key register, LINCALOCK. The LINCALSTA register is a read only register that shows which mode is currently active. This information is held in Register LINCALSTA, Bit 0. Register LINCALSTA, Bits[2:1] can also be used to determine if the LPTRIM value or HPTRIM value has been altered by the system since the LINCALSTA register was last read. The high frequency oscillator can be trimmed in either low or high precision mode (set by Register PWRMOD, Bit 6).

There are two modes of trimming: user trim mode and system trim mode.

User Trim Mode

User trim mode allows any trim register values to be written and downloaded by the user. This mode is always the first trim mode. Factory calculated trim values are automatically contained within trim registers as the default.

The required value is written to the LINCALVAL0 register and the LINCALVAL1 register. In user trim mode, these values automatically match the values in the LINCALVAL2 register and the LINCALVAL3 register. The values in the LINCALVAL2 and LINCALVAL3 read only registers are sent to the oscillator.

The LINCALVAL0 and LINCALVAL1 registers are key protected and require two sequential writes: write the unlock key to the LINCALOCK register followed by the desired trim value to the LINCALVAL0 or LINCALVAL1 register.

The following is an example of how to program a user trim value:

```
LINCALOCK   = LIN_CAL_KEY;     // Unlock key protection
LINCALVAL0  = 0x7B;            // Write trim values 0x7B to trim HFOSC LP Mode
LINCALOCK   = LIN_CAL_KEY;     // Unlock key protection
LINCALVAL1  = 0x200;           // Write trim values 0x200 to trim HFOSC HP Mode
```

It is not advisable to write values to these registers unless it is fully understood the effect such a write has on the oscillator frequency. Invalid values may result in corruption of flash data when written.

System Trim Mode

System trim mode allows a calibration of the clock relative to the LIN baud rate of the system. This mode can be used to achieve greater accuracy from either the normal mode or low power mode. System trim mode cannot operate if there are no LIN communications. The ADuCM300 automatically adapts to download internally calculated trim values to operate within >1% accuracy over the full operating range as long as there is at least 1 LIN transaction for every 10°C of change.

After the user trim mode is set at startup, setting the device into system trim mode overrides the user set trim values with iteratively calculated values derived from LIN communications.

Using the LIN baud rate, the device determines the accuracy of the trim value and automatically increments or decrements a step each time a valid LIN communication occurs within a set window of calibration. The calibration window is defined from values set in the LINCALMINL register and the LINCALMINH register, as well as the LINCALMAXL register and LINCALMAXH register, and from the number of steps defined in the LINCALCON register (key protected). The user sets these values. In system trim mode, LINCALVAL2 and LINCALVAL3 may not match LINCALVAL0 and LINCALVAL1.

The following is a typical sequence for starting system trim mode:

```
LINCALMIN    = EXPECTED_LINBR_VALUE-0x20;    //Define tolerance
LINCALMAX    = EXPECTED_LINBR_VALUE+0x20;    //Define tolerance
LIN_CAL_LOCK = LIN_CAL_KEY;                  // unlock key protection
LINCALCON    = 0x1;                          // Enable LINCAL mode with step size = 1
```

HIGH FREQUENCY OSCILLATOR CALIBRATION MEMORY MAPPED REGISTERS

Table 17. High Frequency Oscillator Calibration Memory Mapped Registers (Base Address 0x40005C00)

Offset	Name	Description	Access	Default
0x0000	LINCALCON	LIN calibration control register	Read/write	0x0000
0x001C	LINCALSTA	System calibration status	Read	0x0000
0x0004	LINCALVAL0	User calibration value (low precision mode)	Read/write	0x0000
0x0008	LINCALVAL1	User calibration value (high precision mode)	Read/write	0x0000
0x0020	LINCALVAL2	System calibration value (low precision mode)	Read	0xXXXX
0x0024	LINCALVAL3	System calibration value (high precision mode)	Read	0xXXXX
0x0014	LINCALMINL	Minimum control window, Bits[15:0]	Read/write	0x0000
0x0018	LINCALMINH	Minimum control window, Bits[18:16]	Read/write	0x0000
0x000C	LINCALMAXL	Maximum control window, Bits[15:0]	Read/write	0x0000
0x0010	LINCALMAXH	Maximum control window, Bits[18:16]	Read/write	0x0000
0x0028	LINCALOCK	Calibration lock register	Read/write	0x0000

LIN Calibration Control Register

Address: 0x40005C00, Reset: 0x0000, Name: LINCALCON

Table 18. LINCALCON Register Bit Descriptions (Key Protected)

Bits	Name	Description
[15:3]	Reserved	Reserved. These bits must be set to 0.
[2:1]	Step	System mode oscillator trim step. 00: Step Size 1 01: Step Size 2 10: Step Size 3 11: Step Size 4
0	CALMODE	Calibration mode. 0: user mode. This setting uses factory calibrated trim values by default. 1: system mode. For high frequency oscillator calibration from a LIN synchronization field.

System Calibration Status Register

Address: 0x40005C1C Reset: 0x0000, Name: LINCALSTA

Table 19. LINCALSTA Register Bit Descriptions

Bits	Name	Description
[15:3]	Reserved	Reserved. These bits must be set to 0.
2	LPACC	This bit allows the user to monitor accuracy in low precision mode. 0: reset during a read operation. 1: set when LINCALVAL2 is altered.
1	HPACC	This bit allows the user to monitor accuracy in high precision mode. 0: reset during a read operation. 1: set when LINCALVAL3 is altered.
0	PWRMODE	High precision oscillator (HPOSC) power mode selected for calibration. 0: high precision and power mode 1: lower precision and power mode

User Calibration Value Register (Low Precision Mode)

Address: 0x40005C04, Reset: 0x0000, Name: LINCALVAL0

Table 20. LINCALVAL0 Register Bit Descriptions (Key Protected)

Bits	Name	Description
[15:9]	Reserved	Reserved. These bits must be set to 0.
[8:0]	LPTRIM	The 9-bit, user trim value used in low precision mode.

User Calibration Value Register (High Precision Mode)

Address: 0x40005C08, Reset: 0x0000, Name: LINCALVAL1

Table 21. LINCALVAL1 Register Bit Descriptions (Key Protected)

Bits	Name	Description
[15:9]	Reserved	Reserved. These bits must be set to 0.
[8:0]	HPTRIM	The 9-bit, user trim value used in high precision mode.

System Calibration Value Register (Low Precision Mode)

Address: 0x40005C20, Reset: 0xXXXX, Name: LINCALVAL2

Table 22. LINCALVAL2 Register Bit Descriptions

Bits	Name	Description
[15:9]	Reserved	Reserved.
[8:0]	LPTRIM	The 9-bit, LIN calibration (LINCAL) trim value used by the high frequency oscillator in low precision 3% mode.

System Calibration Value Register (High Precision Mode)

Address: 0x40005C24, Reset: 0xXXXX, Name: LINCALVAL3

Table 23. LINCALVAL3 Register Bit Descriptions

Bits	Name	Description
[15:9]	Reserved	Reserved.
[8:0]	HPTRIM	The 9-bit, LINCAL trim value used by the high frequency oscillator in high precision 1% mode.

Minimum Control Window Register, Bits[15:0]

0x40005C14, Reset: 0x0000, Name: LINCALMINL

Table 24. LINCALMINL Register Bit Descriptions

Bits	Name	Description
[15:0]	MINSYNC	Minimum tolerance value for the high frequency oscillator system trim

Minimum Control Window Register, Bits[18:16]

Address: 0x40005C18, Reset: 0x0000, Name: LINCALMINH

Table 25. LINCALMINH Register Bit Descriptions

Bits	Name	Description
[15:3]	Reserved	Reserved. These bits must be set to 0.
[2:0]	MINSYNC	Minimum tolerance value for the high frequency oscillator system trim. Bits[18:16] with respect to LINCALMINL.

Maximum Control Window Register, Bits[15:0]

Address: 0x40005C0C, Reset: 0x0000, Name: LINCALMAXL

Table 26. LINCALMAXL Register Bit Descriptions

Bits	Name	Description
[15:0]	MAXSYNC	Maximum tolerance value for the high frequency oscillator system trim

Maximum Control Window Register, Bits[18:16]

Address: 0x40005C10, Reset: 0x0000, Name: LINCALMAXH

Table 27. LINCALMAXH Register Bit Descriptions

Bits	Name	Description
[15:3]	Reserved	Reserved. These bits must be set to 0.
[2:0]	MAXSYNC	Maximum tolerance value for the high frequency oscillator system trim. Bits[18:16] with respect to LINCALMAXL.

Calibration Lock Register

Address: 0x40005C28, Reset: 0x0000, Name: LINCALOCK

Table 28. LINCALOCK Register Bit Descriptions

Bits	Name	Description
[15:0]	Lock	This lock register must be written with the unlock key, 0x1324, immediately before the desired value is written to any key protected register.

LOW FREQUENCY OSCILLATOR CALIBRATION

The accuracy of the 32.768 kHz low frequency oscillator can be improved using the high precision, 16 MHz high frequency oscillator.

The hardware counts the number of high frequency oscillator clocks in a specified number of low frequency oscillator clock periods, and the results are compared. If the count is longer or shorter, the oscillator trim value is incremented or decremented accordingly. The user must manually iterate this procedure until no further increments or decrements are possible, or until the results are outside the maximum and minimum calibration register values.

The dedicated calibration MMRs are shown in Table 29.

At the end of the time base period, the value of EXPUCLK (Register TRMUCTGT, Bits[12:0]) and UCLKCNT (Register TRMUCTCNT, Bits[12:0]) are compared within the tolerance specified by TOLSEL (Register TRMCON, Bit 5). The tolerance can be set at a wider tolerance (32 counts) or a tighter tolerance (16 counts). If the EXPUCLK and UCLKCNT values are within this set tolerance, no further increment or decrement occurs.

Otherwise,

If UCLKCNT > EXPUCLK + 16(1 + TOLSEL), the trim value increments by 1 as the low frequency oscillator runs slow.

If UCLKCNT < EXPUCLK − 16(1 + TOLSEL), the trim value decrements by 1 as the low frequency oscillator runs fast.

TOLSEL is either 0 or 1.

The value for EXPUCLK can be calculated using the following equation:

$$EXPUCLK = MAXLFOSC \times (f_{UCLK}/f_{LFOSC}) \quad (1)$$

where:
MAXLFOSC is the number of low frequency oscillator clocks to count.
f_{UCLK} is the frequency of the UCLK.
f_{LFOSC} is the frequency of the low frequency oscillator

The maximum trim value is factory set. There is no option to modify the maximum trim value. If the user exceeds the maximum trim value, the value reverts to factory settings. The minimum trim value is user programmable. If the high frequency oscillator is disabled, the current trim cycle is immediately aborted, with the trim value unaffected.

LOW FREQUENCY OSCILLATOR CALIBRATION MEMORY MAPPED REGISTERS

Table 29. Low Frequency Oscillator Trim Memory Mapped Registers (Base Address 0x40009C00)

Offset	Name	Description	Access	Default
0x0000	TRMSTA	Low frequency oscillator calibration status register	Read	0x00
0x0004	TRMCON	Low frequency oscillator calibration control register	Read/write	0x00
0x0008	TRMMXC	Maximum calibration value register	Read	0x3F
0x000C	TRMMNC	Minimum calibration value register	Read/write	0x00
0x0010	TRMVAL	Oscillator trim value	Read/write	0xXX
0x0014	TRM32TGT	Low frequency oscillator target count	Read/write	0x00
0x0018	TRM32CNT	Low frequency oscillator current count	Read	0x00
0x001C	TRMUCTGT	UCLK target count	Read/write	0x0000
0x0020	TRMUCTCNT	UCLK current count	Read	0x0000

Low Frequency Oscillator Calibration Status Register

Address: 0x40009C00, Reset: 0x00, Name: TRMSTA

These bits are automatically cleared after they are read.

Table 30. TRMSTA Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved.
5	DECCAL	Last calibration cycle was a decrement (even when minimum is reached or the NOINCDEC bit is set).
4	INCCAL	Last calibration cycle was an increment (even when maximum is reached or the NOINCDEC bit is set).
3	MINCAL	Minimum trim value is reached (interrupt source).
2	MAXCAL	Maximum trim value is reached (interrupt source).
1	UCNTOF	UCLK count overflow (interrupt source).
0	CYCEND	Calibration cycle ended (interrupt source). This bit does not indicate that the low frequency oscillator is fully trimmed, only that a single calibration cycle has completed.

Low Frequency Oscillator Calibration Control Register

Address: 0x40009C04, Reset: 0x00, Name: TRMCON

Table 31. TRMCON Register Bit Descriptions

Bits	Name	Description
7	Reserved	Reserved. This bit must be set to 0.
6	NOINCDEC	0: the trim register is incremented or decremented at the end of a calibration cycle. 1: the trim register is not incremented or decremented at the end of a calibration cycle.
5	TOLSEL	Tolerance select. 0: tolerance of 16. 1: tolerance of 32.
4	Enable	Low frequency oscillator calibration enable. 0: disable the calibration block. 1: enable the calibration block.
3	MNIRQEN	0: disable the minimum trim value interrupt. 1: enable the minimum trim value interrupt.
2	MXIRQEN	0: disable the maximum trim value interrupt. 1: enable the maximum trim value interrupt.
1	UIRQEN	0: disable the UCLK counter overflow interrupt. 1: enable the UCLK counter overflow interrupt.
0	CIRQEN	0: disable the cycle end interrupt. 1: enable the cycle end interrupt.

Maximum Calibration Value Register

Address: 0x40009C08, Reset: 0x3F, Name: TRMMXC

Table 32. TRMMXC Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved.
[5:0]	MAXCAL	Maximum trim value (the value in this register is factory set and not user programmable).

Minimum Calibration Value Register

Address: 0x40009C0C, Reset: 0x00, Name: TRMMNC

Table 33. TRMMNC Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	MINCAL	Minimum trim value (the oscillator trim logic does not decrement below the minimum trim value specified in this register).

Oscillator Trim Value Register

Address: 0x40009C10, Reset: 0xXX, Name: TRMVAL

Table 34. TRMVAL Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	VCOTRIM	Calibration trim value. This defaults to MAXCAL if the value exceeds MAXCAL. During each calibration cycle, this value is increased or decreased by one as necessary.

Low Frequency Oscillator Target Count Register

Address: 0x40009C14, Reset: 0x00, Name: TRM32TGT

Table 35. TRM32TGT Register Bit Descriptions

Bits	Name	Description
[7:3]	Reserved	Reserved. These bits must be set to 0.
[2:0]	MAXLFOSC	The number of low frequency oscillator clocks to count. 0 means disable.

Low Frequency Oscillator Current Count Register

Address: 0x40009C18, Reset: 0x00, Name: TRM32CNT

Table 36. TRM32CNT Register Bit Descriptions

Bits	Name	Description
[7:3]	Reserved	Reserved
[2:0]	LFOSCCNT	Stores the current number of low frequency oscillator clocks

UCLK Target Count Register

Address: 0x40009C1C, Reset: 0x0000, Name: TRMUCTGT

Table 37. TRMUCTGT Register Bit Descriptions

Bits	Name	Description
[15:13]	Reserved	Reserved. These bits must be set to 0.
[12:0]	EXPUCLK	Stores the expected number of high frequency oscillator clocks during the time base.

UCLK Current Count Register

Address: 0x40009C20, Reset: 0x0000, Name: TRMUCTCNT

Table 38. TRMUCTCNT Register Bit Descriptions

Bits	Name	Description
[15:13]	Reserved	Reserved
[12:0]	UCLKCNT	Stores the current number of high frequency oscillator clocks during the time base after the calibration cycle is complete.

KERNEL

The ADuCM300 features a protected, on-chip kernel.

After a reset event, the hardware calculates a cyclic redundancy check (CRC) signature of the kernel memory space and compares the calculated CRC to the CRC signature programmed during the production test to ensure the integrity of kernel. If an error occurs, the hardware prevents kernel execution.

If the CRC signature is correct, the kernel reads back the factory calibrated data from the manufacturing data space and loads the data into the registers of various on-chip peripherals, such as

- Precision oscillator
- Low power oscillator
- 33VDD, DVDD18, AVDD18
- Voltage reference
- ADC0 (offset and gain)
- ADC1 (offset and gain)

The following processor registers and user registers are modified by the kernel and are different from the POR default values:

- R0 to R15
- GP0CON
- FEEADR, FEEDATL, FEEDATH, FEECON0, FEECON1, FEESIGN
- HVDAT, HVCON
- HVDCFG0

The ADuCM300 features an on-chip LIN downloader. Downloader exit and kernel entry can only occur via a reset. During kernel execution, SWD access is disabled. Before exiting to user code, the kernel checks if the SWD lock location of the uppermost page in flash contains the key value, 0x160320 (see Figure 8 for more information). If this key is present, SWD access is not granted after kernel exit. If any other value is present, SWD access is enabled. This process provides additional security to ensure that SWD access is not possible between kernel exit and user code SWD disabling.

KERNEL IMPLEMENTATION

After a reset, the kernel initializes the device and programs calibration values into peripheral registers.

The kernel then performs a SRAM integrity check. If the SRAM is corrupted, the kernel clears the SRAM before executing a software reset.

The SRAM integrity check handles only 2-bit ECC errors. The user must handle 1-bit ECC errors after the kernel exit.

After the SRAM check, the kernel determines which mode of operation it must branch to. This mode is either debug mode, user mode, bootloader mode, or LIN download mode.

Debug Mode

After a reset, the kernel enters debug mode when the input level of the GPIO5 pin is low. The kernel branches to user space using Address 0x00000000. This mode of operation is only intended for code development because it bypasses the page 0 corruption check.

The user ensures the input level of the GPIO5 pin is low prior to a reset being issued to ensure entering debug mode.

Normal Kernel Operating Mode

In normal kernel operating mode, if after a reset event, the kernel finds the input level of the GPIO5 pin high, the kernel checks for a valid first page of user flash. A valid page is one that has a valid CRC signature at Address 0x7FC. If the kernel determines that the first page of user flash is valid, the kernel branches to the beginning of that page. For robustness, the first page must contain code that validates the rest of the application code before exiting the first page.

A default key value of 0x16400000 can be used instead of the CRC at Address 0x7FC. If this key is found at Address 0x7FC, the kernel branches to the beginning of Page 0. However, using the default key does not ensure that the first page is not corrupted.

If the 0x7FC location has already been programmed and user code is required to modify it, either Page 0 must be erased or the 0x7FC location must be rewritten with all zeros. Overwriting the 0x7FC location with any other value is not recommended because an invalid ECC can result.

Bootloader Mode

If, after a reset, the kernel finds the input level of the GPIO5 pin to be high and Page 0 not to have a valid CRC or default key, the kernel checks for a valid user bootloader. The bootloader can be of any size up to 30 kB, but must be located at the top of the user flash and contain the following basic bootloader data:

- Address (0x20000, −0x4) must contain the CRC of the bootloader.
- Address (0x20000, −0x1C) must contain the lowest address of the bootloader block.
- Address (0x20000, −0x20) must contain the entry point of the bootloader code.

All three addresses must contain valid information to enter the bootloader properly.

The user flash size for the ADuCM300 (128 kB) is 0x20000 bytes.

Due to the Cortex-M3 addressing architecture, any address branched to must be a half word boundary + 1. For example, if the lowest address of the bootloader on a 128 kb device (ADuCM300) is 0x18800, the entry point must be 0x18801.

The kernel uses the basic bootloader data (CRC, lowest address, and entry point) to determine if the bootloader is valid. If the bootloader is not valid, the kernel enters download mode and waits for data via the LIN interface.

In user application mode, the device can receive commands via the LIN interface to enter the bootloader mode, which is located at the top of the user flash. The bootloader, using the appropriate user protocol, can update the application code. Before entering the bootloader, the user must ensure that the value at Address 0x7FC is not the CRC for Page 0 or the key value (0x16400000). The invalid value at Address 0x7FC ensures that Page 0 appears invalid (see Figure 4). During the last step of programming the flash, Address 0x7FC must be written with either the CRC of Page 0 or the key value to ensure the kernel can enter user application mode.

If the bootloader locations have already been programmed and user code is required to modify them, the user must either erase that page or overwrite the location with all zeros. Overwriting the bootloader locations with any other value is not recommended because an invalid ECC can result.

If the bootloader feature is not used, the entry point of the bootloader, add flash address (0x20000 - 0x20) must be set to a value of 0xFFFFFFFF. This prevents unwanted attempts to enter the bootloader function.

Interrupted Bootloading

If the user defined flash programming procedure (bootloading) is interrupted before completion, the bootloading can be restarted after a reset is executed. Following a reset, after a partial programming of the flash, the kernel detects that the application is corrupted (value at Address 0x7FC is incorrect) and, as shown in Figure 4, checks for a valid bootloader. The kernel passes control to the bootloader and the user defined flash programming procedure is repeated.

Downloader Mode

If the kernel finds that the bootloader is not valid, the kernel enters the basic downloader of the kernel, and it is possible to program the flash with user code as described in the AN-946 Application Note. When in this state, if no valid LIN frames are received within about one hour, the device enters a fail safe mode. This mode can only be exited with a full power cycle.

The download mode supports a fast LIN download option enabling the user to program the device at speeds of up to 100 kbaud.

The download mode cannot be entered from any of the other modes except via a reset.

After user code has been downloaded via LIN, a reset is required for the kernel to enter user mode.

Interrupted Downloading

If downloading to the user space is interrupted before the download is complete, the download can be restarted as long as Address 0x7FC does not contain the Page 0 CRC or the key (0x16400000) and the bootloader is not valid. The downloader can be restarted at any time by resetting the device. To allow restarting of download, Address 0x7FC of Page 0 must only be written to at the end of the download process. In addition, if the bootloader feature is used, the values at-Address 0x20000−0x4 and Address 0x20000−0x1C must be updated only after the entire bootloader has been downloaded and verified.

Direct LIN Interface (External Transceiver Mode)

When entering the download mode, the kernel drives the GPIO3 pin high, which can be used to enable an external LIN interface. The kernel then monitors both the LIN pin and the Rx pin. If a frame start is detected on the LIN pin, the kernel assumes internal LIN transceiver is used to download the user code. If a frame start is first detected on the LIN_RX pin (GPIO4), the kernel switches over to use the LIN_RX/LIN_TX pins (GPIO4/GPIO1) instead. This allows use of an external LIN transceiver. For the kernel, the only difference between these modes is this switching from the LIN pin to the LIN_RX/LIN_TX pins. The kernel only performs this switching when it

reaches download mode. If the kernel exits to user or bootloader mode, the user code must switch to the LIN_RX/LIN_TX pins and drive the GPIO3 pin to control the transceiver.

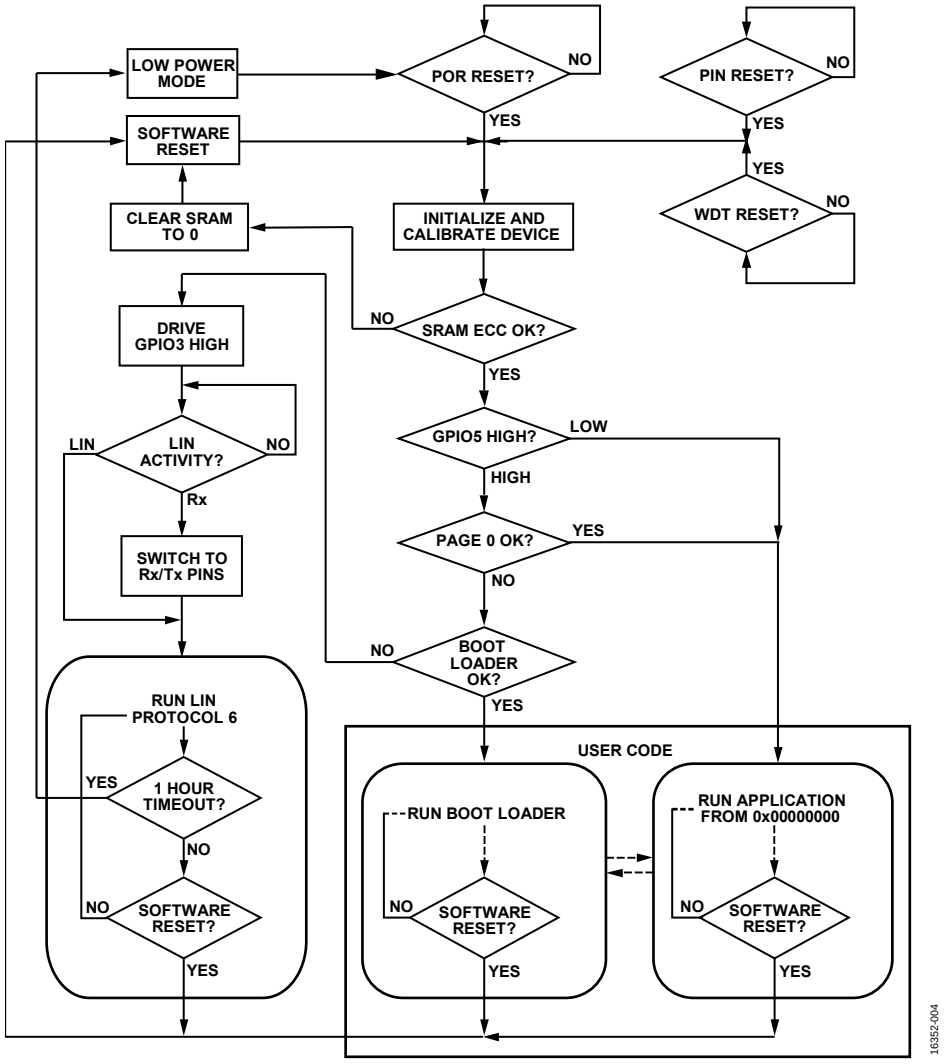

Figure 4. Kernel Flowchart

RESET

RESET FEATURES

There are four kinds of resets:

- Software system reset (SWR)
- External reset (EXR)
- Power on reset (POR)
- Watchdog timeout reset (WDR)

RESET OPERATION

The SWR is provided as part of the Cortex-M3 core and allows the device to be put into a known state. To generate an SWR, the value of 0x05FA0004 must be written to the application interrupt and reset control register (AIRCR register). This register is part of the NVIC and is located at Address 0xE000ED0C. The RSTSTA register stores the cause for the reset until the RSTSTA register is cleared by writing to the RSTCLR register. RSTSTA and RSTCLR can be used during a reset exception service routine to identify the source of the reset.

The $\overline{\text{RESET}}$ pin does not reset the debug logic.

Table 39. Device Reset Implications

Reset	Reset External Pins to Default State	Execute Kernel	Reset All MMRs Except RSTSTA	Reset All Top Die Registers	Reset All Peripherals	Valid SRAM	RSTSTA After Reset Event
SWR	Yes	Yes	Yes	No	Yes	RAM is not valid when an ECC error is detected during kernel initialization (SRAM initialized to zero).	Register RSTSTA, Bit 3 = 1
WDR	Yes	Yes	Yes	No	Yes	RAM is not valid when an ECC error is detected during kernel initialization (SRAM initialized to zero).	Register RSTSTA, Bit 2 = 1
EXR	Yes	Yes	Yes	No	Yes	RAM is not valid when an ECC error is detected during kernel initialization (SRAM initialized to zero).	Register RSTSTA, Bit 1 = 1
POR	Yes	Yes	Yes	Yes	Yes	RAM is not valid	Register RSTSTA, Bit 0 = 1

RESET MEMORY MAPPED REGISTER

Table 40. Reset Memory Mapped Register (Base Address 0x40002400)

Offset	Name	Description	Access	Default
0x0040	RSTSTA	Reset status register	Read	Depends on the type of reset
0x0040	RSTCLR	Reset clear register	Write	Not applicable

Reset Status Register and Reset Clear Register: RSTSTA and RSTCLR

Address: 0x40002440, Reset: dependent on type of reset, Name: RSTSTA

Address: 0x40002440, Reset: not applicable, Name: RSTCLR

Table 41. RSTSTA and RSTCLR Register Bit Descriptions

Bits	Name	Description
[7:4]	Reserved	Reserved
3	SWRST	Software reset 0: cleared by setting the corresponding bit in the RSTCLR register 1: set automatically when the Cortex-M3 system reset is generated
2	WDRST	Watchdog timeout 0: cleared by setting the corresponding bit in the RSTCLR register 1: set automatically when a watchdog timeout occurs
1	EXTRST	External reset 0: cleared by setting the corresponding bit in the RSTCLR register 1: set automatically when an external reset occurs

Bits	Name	Description
0	POR	Power-on reset 0: cleared by setting the corresponding bit in the RSTCLR register 1: set automatically when a power-on reset occurs

The bits in the RSTSTA and RSTCLR register must be checked as part of the power-up self check. The user must poll these bits after any reset and check for consistency. An unexpected reset event must be treated as a potential fault condition. Take appropriate action to put the device into a safe state.

MEMORY ORGANIZATION

Three separate blocks of memory are accessible to the user, and they are as follows:

- 6 kB of SRAM from 0x20000000 to 0x200017FF
- 4 kB of data flash memory 0x00400000 to 0x00400FFF
- 128 kB of on-chip Flash/EE memory available to the user from 0x00000000 to 0x0001FFFF

There is also an additional 2 kB reserved for the kernel space from 0x00020000 to 0x000207FF.

These blocks are mapped according to the Cortex-M3 memory map, as shown in Figure 5. All on-chip peripherals are accessed via the MMRs, situated in the bit band region. Any access to MMRs takes three clock cycles of the clock used in the related functional block, unless otherwise noted.

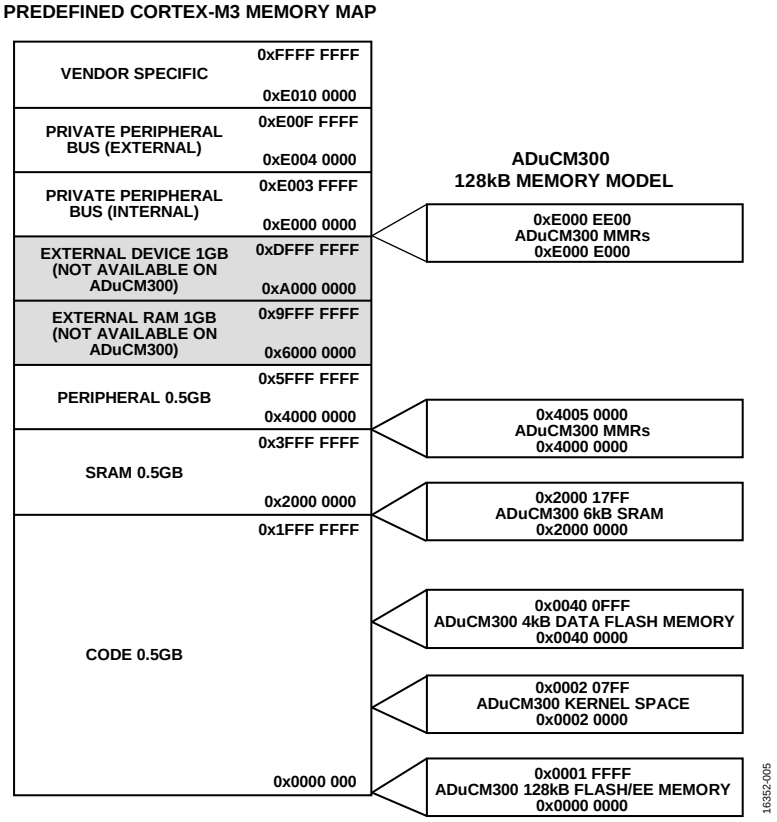

Figure 5. Memory Map Diagram

FLASH CONTROLLER

FLASH CONTROLLER FEATURES

The flash memory available to the user are 128 kB program flash and 4 kB data flash.

Commands Supported

The flash memory command options available to the user are write, mass erase, page erase, generation of signatures for single pages or multiple pages, and command abort.

Accesses from the core on program flash are not stalled if the command in progress is in data flash. Accesses in data flash are stalled if the command in progress is in program flash.

Flash Protection

The flash memory protection options available to the user are write protection for user space and data flash and the ability to lock serial wire interface.

Flash Integrity

The flash memory integrity check options available to the user are

- Automatic signature check of kernel space on reset.
- User signature for application code.
- 8-bit ECC includes a 1-bit error correction, which can generate an interrupt, and a 2-bit detection, which generates a hard fault exception.

FLASH CONTROLLER OVERVIEW

The flash controller supports two embedded high data retention (HDR) flash memories: 128 kB (ADuCM300) program flash and 4 kB data flash. Program flash memory is for storing user code and has an additional 2 kB of information space to store the kernel. Data flash memory can store additional data by the user. A write to the flash is executed via keyhole access.

FLASH MEMORY ORGANIZATION

On the ADuCM300, the controller supports 128 kB of program flash ending at Address 0x1FFFF with 2 kB of information space containing the kernel, as shown in Figure 6.

The ADuCM300 additionally contains a separate block with 4 kB of data flash memory, as shown in Figure 7. Page sizes are 2 kB for program flash and 512 bytes for data flash.

Program Flash Information Space

The program flash information space is mapped above the program flash user space. The information space contains the kernel and default calibration and configuration data.

Program Flash User Space

The top 24 bytes of program flash user space are reserved for a signature, the user write protection (program flash write protection), and the user failure analysis key (USERFAKEYx) registers, as shown in Figure 8.

If the user tries to read from or write to a portion of memory that is not available, a bus error is returned. If the user tries to write via the keyhole to a portion of memory that is not available, the appropriate error flag is set.

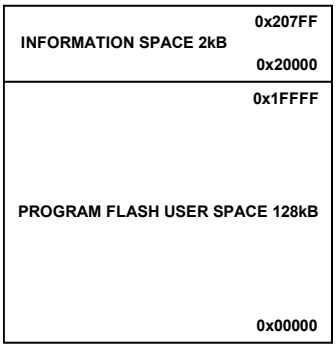

Figure 6. Program Flash Memory Map

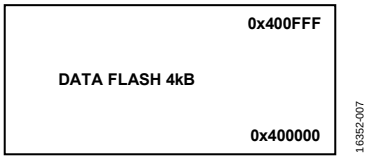

Figure 7. Data Flash Memory Map

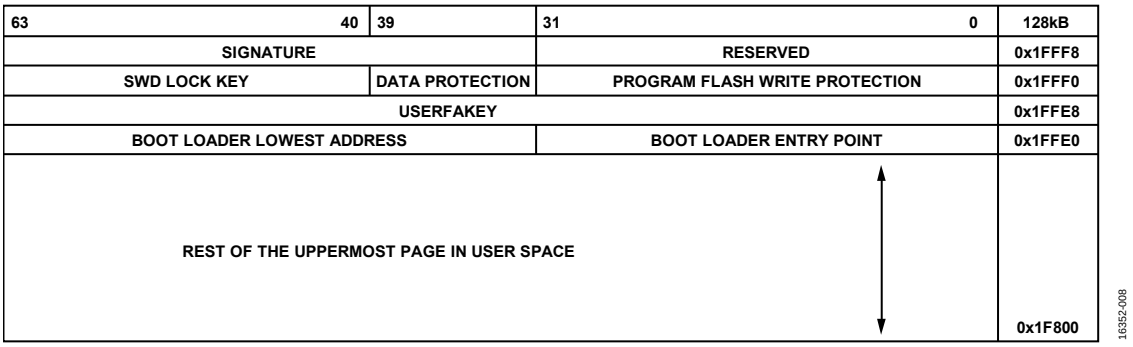

Figure 8. Uppermost Page of User Memory

WRITING TO FLASH/EE MEMORY

Writing to program and data flash is achieved through keyhole access. Each write programs 64 bits of data.

Keyhole access consists of flash address, data registers, a key register, and a command register.

To write to a flash location, the following sequence is required:

1. Write to the FEEADDR register with the 23-bit memory mapped address of the flash location.
2. Write to the FEEDATL register and the FEEDATH register with 64 bits of data.
3. Write 0xF456 followed by 0xF123 to the FEEKEY register.
4. Write to the FEECMD register with write command.

After the write command is given, the controller writes to the program or the data flash based on the address provided. A single 64-bit location can only be written to once without an erase.

Flash Memory Write Code Example

```
unsigned int uiSTA;
FEEADR  = 0x00001800;         // A 64-bit flash location
FEEDATL = 0x01234567;         // Data
FEEDATH = 0x89ABCDEF;
FEEKEY  = 0xF456;             // Enter Key
FEEKEY  = 0xF123;
FEECMD  = 0x4;                // Flash write command
do{
  do{
    uiSTA = FEESTA;
  }while(uiSTA & 0x1);         // Wait until not busy
}while( ! (uiSTA & 0x4));      // Ensure command completed
```

ERASING FLASH/EE MEMORY

User code can call the following two flash erase commands:

- Mass erase: this command erases the entire user flash memory. After entering the user protection key into the FEEKEY register, write the mass erase command to the FEECMD register. Example code is provided in the Mass Erase of Data Flash Code Example section.
- Page erase: this command erases 2 kB in user space of program flash or 512 bytes of data flash. The page is selected by the FEEADR1L register. After entering the user protection keys into the FEEKEY register, load the FEEADR1L register with the page address to be erased. Finally, write the page erase command to the FEECMD register. CMDDONE (Register FEESTA, Bit 2) indicates that the page erase command is complete. Example code is provided in the Page Erase Code Example section.

During a page erase or mass erase sequence, the flash controller and flash block consume extra current for the duration of the flash erase sequence. See the ADuCM300 data sheet for exact specifications.

Mass Erase of Data Flash Code Example

```
unsigned int uiSTA;

FEEKEY = 0xF456;                    // Enter Keys
FEEKEY = 0xF123;
FEECMD = 0x6;                       // Mass Erase Data Flash

do{
  do{
    uiSTA = FEESTA;
  }while(uiSTA & 0x1);              // Wait until not busy
}while( ! (uiSTA & 0x4));           // Ensure command completed
```

Page Erase Code Example

```
unsigned int uiSTA;

FEEADR1L = 0x00010000;                // A location in the page
                                      // Example uses Page 32
FEEKEY   = 0xF456;                    // Enter Keys
FEEKEY   = 0xF123;
FEECMD   = 0x1;                       // Flash erase page command

do{
  do{
    uiSTA = FEESTA;
  }while(uiSTA & 0x1);                // Wait until not busy
}while( ! (uiSTA & 0x4));             // Ensure command completed
```

FLASH CONTROLLER OPERATION

The ADuCM300 flash controller supports simultaneous access to both data and program flash during certain operations. Table 42 shows these operations.

Table 42. Flash Controller Access Matrix

Data Flash	Program Flash	Available
Standby	Standby	Yes
Read	Standby	Yes
Program	Standby	Yes
Erase	Standby	Yes
Standby	Read	Yes
Standby	Program	Yes
Standby	Erase	Yes
Read	Read	Yes
Program	Read	Yes
Erase	Read	Yes
Read	Program	No
Read	Erase	No
Program/Erase	Program	No
Erase/Program	Erase	No

FLASH PROTECTION

The three types of protection that are implemented include key protection, read protection, and write protection.

Flash Protection: Key Protection

Some of the flash controller registers are key protected to avoid accidental writes to these registers.

The user key consists of two writes to the FEEKEY register: 0xF456 followed by 0xF123 are written to the register to enter the key. The 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 (FEECON1). When entered, the key remains asserted unless a command is written to the FEECMD register. When the command starts, the key clears automatically.

If, for example, the key is entered to write to certain locations in flash, and the user write procedure is interrupted before the command is entered, ensure that the key is cleared by writing any value to the key register to prevent an unintended write.

Flash Protection: User Read Protection

User space read protection is provided by disabling serial wire access. The user can disable serial wire access by writing 0 to the DBG bit in the flash FEECON1 register. Serial wire access is disabled while the kernel is running. Otherwise, serial wire access can prevent the kernel from running to completion. When the kernel has completed, it reenables serial wire access for the user (unless the SWD restriction key is valid). See the Kernel section for more information.

Flash Protection: 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, the ADuCM300 flash controller generates an interrupt when the command error or complete interrupt is enabled.

The user program flash write protection is located in the uppermost page of user memory space. Refer to Figure 8 for details.

Table 43. 128 kB Program Flash Write Protection, Bits[31:16]

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Page 62 and Page 63	Page 60 and Page 61	Page 58 and Page 59	Page 56 and Page 57	Page 54 and Page 55	Page 52 and Page 53	Page 50 and Page 51	Page 48 and Page 49	Page 46 and Page 47	Page 44 and Page 45	Page 42 and Page 43	Page 40 and Page 41	Page 38 and Page 39	Page 36 and Page 37	Page 34 and Page 35	Page 32 and Page 33

Table 44. 128 kB Program Flash Write Protection, Bits[15:0]

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Page 30 and Page 31	Page 28 and Page 29	Page 26 and Page 27	Page 24 and Page 25	Page 22 and Page 23	Page 20 and Page 21	Page 18 and Page 19	Page 16 and Page 17	Page 14 and Page 15	Page 12 and Page 13	Page 10 and Page 11	Page 8 and Page 9	Page 6 and Page 7	Page 4 and Page 5	Page 2 and Page 3	Page 0 and Page 1

Table 45. SWD Lock Key, Bits[63:40], 4 kB Data Flash Write Protection, Bits[39:32]

Bits[63:40]	Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit 32
SWD Lock Key	Page 7	Page 6	Page 5	Page 4	Page 3	Page 2	Page 1	Page 0

The flash controller uploads the write protection into local registers after a reset. To write to the write protection bits, the user must first write 0xF456 followed by 0xF123 to the key register. After the write protection is written, the write protection cannot be rewritten without a mass erase of user space or a page erase of the last page (if the last page is not protected). After a mass erase, the device must be reset to deassert the uploaded copy of the write protection bits.

The following is the sequence to program the write protection, Bits[31:0]:

1. Ensure that the last page of user space is erased.
2. Write 0xF456 followed by 0xF123 to the key register, FEEKEY.
3. Write the required write protection directly to flash. Write 0 to enable protection. The write protection address is 0x1FFF0 for 128 kB of flash.
4. Verify that the write completed successfully by polling the status register, FEESTA, Bit 3, or by enabling a write complete interrupt.
5. Reset the device, and the write protection is uploaded from the user space and activated by the flash controller.

If the write protection in flash has not been programmed (that is, 0xFFFFFFFF is uploaded from flash on power-up), the FEEPROP register and the FEEPROD register can be written to directly from user code. Writing to the registers directly from user code allows the user to verify the write protection before committing it to flash. If the write protection in flash has been programmed, the MSB of the write protection must be programmed to 0 to prevent erasing of the write protection block. For the ADuCM300 write protection, the memory is split into 32 blocks. For 128 kB of flash, memory is split into 32 × 4 kB blocks.

If an attempt is made to write to the write protection word in flash without setting the FEEKEY register first, a flag is set.

FLASH CONTROLLER FAILURE ANALYSIS KEY

It can be necessary to perform failure analysis on devices that are returned by the user even though read protection is enabled. A method is provided to allow failure analysis of protected memory by a user failure analysis (FA) key.

The user FA key is a 64-bit key that is stored at the top of the user space in flash, as shown in Figure 8. This key is used to gain access to user code if the serial wire interface is locked. It is the responsibility of the user to program this key to a value. A value of 0xFFFFFFFF is the default, and any other value programmed in this location is treated as the user FA key. This same key must be programmed into the USERFAKEYx register to unlock read protection via SWD. The key must be given to Analog Devices to enable access to user code.

FLASH INTEGRITY SIGNATURE FEATURE

The signature is used to check the integrity of the flash device. The signature is calculated on 64-bit data by splitting it into two 32-bit data-words. The CRC calculation sequence is first the lower 32 bits, then the higher 32 bits of the 64-bit flash data.

The ECC is checked on each flash read. If errors are corrected by the ECC, the ERRCORRECTED flag in the status register is set after the signature check is completed and an interrupt is raised. If errors are detected and cannot be corrected by ECC, the ERRDETECTED flag in the status register is set and an interrupt is raised. A signature check is treated as a failure when the computed signature is not equal to the stored signature. The software can call a signature check command occasionally or whenever a new block of code is about to be executed. The signature is a 24-bit CRC with the polynomial $x^{24} + x^{23} + x^6 + x^5 + x + 1$. Contact Analog Devices for more information, if required.

The sign command can be used to generate or check the signature of a block of code, where a block can be a single page or multiple pages. A 24-bit linear feedback shift register is used to generate the signature. The hardware assumes that the signature for a block is stored in the upper four bytes of the most significant page of a block. Therefore, these four bytes cannot be included when generating the signature.

The following procedure must be followed to generate a signature:

1. Write the start address of the block to the FEEADR1L register.
2. Write the end address of the block to the FEEADR1H register.
3. Write the sign command to the command register.
4. When the command is complete, the signature is available in the flash signature register (FEESIGN). The signature is compared with the data stored in the upper four bytes of the uppermost page of the block. If the data does not match the signature, a fail status of VERIFYERR is returned in the status register (Register FEESTA, Bits[5:4] = 10).

While the signature is being computed, all other accesses to flash are stalled.

The user must run the CRC polynomial in user code first to generate the CRC value and then must write the generated CRC value to the upper four bytes of the uppermost page of a block. When this operation is complete, any call of the signature feature compares this 4-byte value to the result of the signature check function.

ECC Error Handling

During a read of the flash, if there is a 1-bit error, the error is corrected, and the appropriate flags are set in the status register. A 1-bit ECC interrupt in the command control register (FEECON0) must be enabled.

If there is a 2-bit ECC error, an error is issued by the controller. ECC errors that are two bits or greater generate a bus exception, unless they are encountered by the sign command.

An ECC error is signaled by the ECC error detection/correction module when a flash location is read. Depending on when the read occurs (for example, during command execution or during a read) and from which flash (program or data) the read occurs, the appropriate flags are set in the status register (for example, ECCERRCMD or ECCERRREADxx).

ECC Error During Read

Because a program and data flash read can happen simultaneously, two separate ECCERRREADxx flags are present in the status register. The flags are Register FEESTA, Bits[10:9] and Register FEESTA, Bits[12:11]. If the interrupt is configured to be generated when an ECC error occurs, the address at which the error is detected is available for readback to the user.

ECC Error During Execution of Sign Command

If there is an ECC error during signature check, ECC error data and address registers are not updated. After the command is complete, the ECCERRCMD flags in Register FEESTA, Bits[8:7] are updated. If ECCCMDINTEN in the FEECON0 register is set and any of the ECCERRCMD bits are set, an interrupt is generated.

FLASH CONTROLLER PERFORMANCE AND COMMAND DURATION

Typical command duration times are as follows:

- Direct single write access (72-bit location): 72.187 µs
- Mass erase: 17.126 ms
- Page erase: 17.012 ms
- Page write: program flash user space (256, 72-bit locations) = 8 × 72.187 µs + 248 × 49.7 4 µs = 12.913 ms

For a 4 kB data flash, each row has 32 × 72-bit locations. However, there are only two rows per page. Assuming that the page is written in sequence, and that the writes are done back to back, the following is true for a page write:

4 kB data flash = 2 × ((1 × 72.187 µs) + (31 × 49.74 µs)) = 3.228 ms

FLASH CONTROLLER MEMORY MAPPED REGISTERS

Table 46. Flash Controller Memory Mapped Registers (Base Address 0x40018000)

Offset	Name	Description	Access	Default
0x0000	FEESTA	Flash memory status register	Read	0x00000000
0x0004	FEECON0	Flash memory command control register	Read/write	0x0010
0x0008	FEECMD	Flash memory command register	Read/write	0x0000
0x000C	FEEADR	Flash address keyhole register	Read/write	0x00000000
0x0010	FEEDATL	Flash data register (lower)	Read/write	0x00000000
0x0014	FEEDATH	Flash data register (upper)	Read/write	0x00000000
0x0018	FEEADR1L	Flash controller lower page address register	Read/write	0x00000000
0x001C	FEEADR1H	Flash controller upper page address register	Read/write	0x00000000
0x0020	FEEKEY	Flash controller key register	Write	0x0000
0x0028	FEEPROP	Program flash write protection register	Read/write	0xFFFFFFFF
0x002C	FEEPROD	Data flash write protection register	Read/write	0xFF
0x0030	FEEECC	Data flash ECC disable register	Read/write	0x00000000
0x0034	FEESIGN	Flash controller signature register	Read	0x00000000
0x0038	FEECON1	Serial wire control register	Read/write	0x0001
0x0040	FEEABORT	Flash controller write abort address register	Read	0x00000000
0x0048	FEEAEN0	Flash controller abort enable register	Read/write	0x0000
0x0068	USERFAKEY0	USERFAKEY low register, Bits[31:0]	Read/write	0x00000000
0x006C	USERFAKEY1	USERFAKEY high register, Bits[63:32]	Read/write	0x00000000
0x0074	FEEPECC	Program flash address for ECC error	Read	0x00000000
0x0078	FEEDECC	Data flash address for ECC error	Read	0x00000000

Flash Memory Status Register

Address: 0x40018000, Reset: 0x00000000, Name: FEESTA

Table 47. FEESTA Register Bit Descriptions

Bits	Name	Description
[31:25]	Reserved	Reserved.
[24:22]	ECCCOUNTDATA	This is a 3-bit counter that reflects the number of 1-bit ECC read errors in data flash after Register FEESTA, Bits[12:11] = 0x2 and before FEESTA is read. This counter does not count ECC 2-bit errors. The counter is cleared when FEESTA is read by the user.
[21:20]	Reserved	Reserved.
[19:17]	ECCCOUNTPROG	This is a 3-bit counter that reflects the number of 1-bit ECC read errors in program flash after Register FEESTA, Bits[10:9] = 0x2 and before FEESTA is read. This counter does not count ECC 2-bit errors. The counter is cleared when FEESTA is read by the user.
[16:15]	ECCERRSIGN	ECC error during initial signature check. 00: NOERR. No error. Completed flash read operation during initial signature check or page signature check. 01: ERRDETECTED. During initial signature check, 2-bit errors are detected, and not corrected for at least one flash location. 10: ERRCORRECTED. 1-bit error is corrected for one flash location during a signature command. 11: ERR1BIT_2BIT. During the initial signature command, 1-bit errors and 2-bit errors are detected on one or more flash locations.
14	INIT	Initialization upload in progress. After a reset, the flash controller uploads the flash configuration from the information space, checks the information space signature, and uploads the user write protection. 0: cleared to 0 when the upload completes. User code cannot run until this bit deasserts. 1: set to 1 while the upload is in progress.
13	SIGNERR	Information space signature check on reset error. After a reset, the flash controller automatically checks the information space signature. User code does not execute if this bit is set. 0: cleared to 0 if the signature check returns no errors. 1: set to 1 if the signature check fails.
[12:11]	ECCERRREADDT	ECC errors during a read of data flash. 00: NOERR. No error. Completed read from data flash. 01: ERRDETECTED. 2-bit error detected in one or more flash locations during a read from data flash. The errors are not corrected. 10: ERRCORRECTED. 1-bit error detected for one flash location during a read from data flash. The error is corrected. 11: ERR1BIT_2Bit. During the initial signature command, 1-bit errors and 2-bit errors are detected on one or more flash locations.
[10:9]	ECCERRREADPG	ECC errors during read of program flash. 00: NOERR. No error. Completed read from program flash.
		01: ERRDETECTED. 2-bit error detected in one or more flash locations during a read from program flash. The errors are not corrected. 10: ERRCORRECTED. 1-bit error detected for one flash location during read from program flash. The error is corrected. 11: ERR1BIT_2Bit. During the read, 1-bit errors and 2-bit errors are detected in program flash.
[8:7]	ECCERRCMD	ECC errors. 00: NOERR. No error. Completed flash read operation during the signature check. 01: ERRDETECTED. 2-bit error detected in one or more flash locations during the signature command. The errors are not corrected. 10: ERRCORRECTED. 1-bit error detected for one flash location while performing a signature check. The error is corrected. 11: ERR1BIT_2Bit. During the signature command, 1-bit errors and 2-bit errors are detected on one or more flash locations.
6	Reserved	Reserved.

Bits	Name	Description
[5:4]	CMDFAIL	Status of a command on completion. 00: SUCCESSCOMP. Successful completion of a command or a write. 01: LOCATIONPROT. Attempted write or erase of a protected location. The command is ignored. 10: VERIFYERR. Read verify error. After an erase, the controller reads the corresponding word(s) to verify that the transaction completed. If the data read is not all Fs (FFFFFFFF), this is the resulting status. If the sign command is executed, and the resulting signature does not match the data in the upper four bytes of the upper page in a block, this is the resulting status. 11: aborted. Indicates that a command or a write was aborted by an abort command, or that a system interrupt caused an abort.
3	WRALCOMP	Write almost complete. 0: cleared when read. 1: set to 1 after the second 24-bit write of the three 24-bit writes is complete.
2	CMDDONE	Command complete. 0: cleared when read. 1: set to 1 when a command completes. If there are multiple commands, this status bit asserts after the first command completes and stays asserted until read. It is recommended to wait until this bit is set before continuing.
1	WRCLOSE	Write close. 0: cleared after the WRALCOMP bit is set to 1. 1: set to 1 when the user writes all keyhole registers for a flash write, and the controller starts writing. If this bit is set to 1, all keyhole registers except the command register are closed for writing.
0	CMDBUSY	Command busy. 0: cleared to 0 when the flash block is not executing any commands entered via the command register. 1: set to 1 when the flash block is executing a command entered via the flash memory command (FEECMD) register.

Flash Memory Command Control Register

Address: 0x40018004, Reset: 0x0010, Name: FEECON0

Table 48. FEECON0 Register Bit Descriptions

Bits	Name	Description
[15:5]	Reserved	Reserved. These bits must be set to 0.
4	ECCCMDINTEN	Interrupt enable when a 1-bit ECC error occurs during a read from program or data flash. This bit is enabled by default.
3	Reserved	Reserved. This bit must be set to 0.
2	CMDERRINTEN	Command fail interrupt enable.
1	WRALCOMP	Write almost complete interrupt enable.
0	CMDCOMPINTEN	Command complete interrupt enable.

Flash Memory Command Register

Address: 0x40018008, Reset: 0x0000, Name: FEECMD

Table 49. FEECMD Register Bit Descriptions

Bits	Name	Description
[15:4]	Reserved	Reserved. These bits must be set to 0.
[3:0]	CMD	Flash controller commands (see Table 50).

The commands listed in Table 50 are supported by the flash block. For repeated page erase commands, the key must be entered before each command. If a command is entered without entering the key first, no action is taken, and CMDDONE does not assert. Accesses from the core on program flash are not stalled if the command in progress is in data flash. Accesses in data flash are stalled if the command in progress is in program flash. CMDDONE is asserted if the key is not entered or if an incorrect key is entered.

Table 50. Flash Controller Commands (Register FEECMD, Bits[3:0])

Command	Name	Description
0000	Idle	No command executed.
0001	ERASEPAGE	Write the address of the page to be erased to the FEEADR1L 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 that all the words in the page are erased. If there is a read verify error, the read verify error is indicated in the FEESTA register. ECC is disabled for this command. 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.
0010	Sign	Use this command to generate a signature for a block of data. The signature is generated on a page by page basis. To generate a signature, enter the address of the first page of the block in FEEADR1L, write the address of the last page to FEEADR1H, then write this code to the FEECMD register. When the command is complete, the signature is available for reading in the FEESIGN register. The last four bytes of the last page in a block is reserved for storing the signature. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register. ECC is checked with this command (even if it is specifically disabled for a page). ECC errors produced during the sign command update Register FEESTA, Bits[8:7]. ECC errors produced during the sign command do not update the FEEPECC or FEEDECC registers. ECC errors during the sign command only generate an interrupt if Register FEECON0, Bit 2 is set to 1.
0100	Write	Write to flash locations. This command takes the address from the FEEADR register and data from the FEEDATx keyhole registers. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register before writing into the write protection location and the user FA key location. No key is required for other flash locations.
0101	MASSERASEPROG	Erase all of user space in program flash. 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 FEESTA register. ECC is disabled for this command. To enable this operation, 0xF456 followed by 0xF123 must be written to the FEEKEY register. This write is to prevent accidental erases.
0110	MASSERASEDATA	Erase the data flash (only the first 2048 bytes). When the mass erase is complete, the controller reads every location of data flash to verify that all locations are 0xFFFFFFFFFFFFFFFF. If there is a read verify error, it is indicated in the status register. ECC is disabled for this command. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register.
0111	MASSERASEALLDATA	Erase all user space and all information space in data flash (4096 bytes). When the mass erase is complete, the controller reads every location of data flash to verify that all locations are 0xFFFFFFFFFFFFFFFF. If there is a read verify error, it is indicated in the status register. ECC is disabled for this command. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register.
1000	Abort	If this command is issued, any command currently in progress is stopped. The status indicates the command is completed with an error status in Register FEESTA, Bits[6:4]. 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 is in progress. 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 FEEKEY register to prevent accidental aborts.

Flash Address Keyhole Register

Address: 0x4001800C, Reset: 0x00000000, Name: FEEADR

Table 51. FEEADR Register Bit Descriptions

Bits	Name	Description
[31:23]	Reserved	Reserved
[22:3]	Value	Memory mapped address for the flash location
[2:0]	Reserved	Reserved

Flash Data Register (Lower)

Address: 0x40018010, Reset: 0xF300130F, Name: FEEDATL

Table 52. FEEDATL Register Bit Descriptions

Bits	Name	Description
[31:0]	Value	Bits[31:0] of the data to be written to flash. After a reset, this register is modified by the kernel and holds device traceability information.

Flash Data Register (Upper)

Address: 0x40018014, Reset: 0x00000000, Name: FEEDATH

Table 53. FEEDATH Register Bit Descriptions

Bits	Name	Description
[31:0]	Value	Bits[63:32] of the data to be written to flash.

Flash Controller Lower Page Address Register

Address: 0x40018018, Reset: 0x00000000, Name: FEEADR1L

Table 54. FEEADR1L Register Bit Descriptions

Bits	Name	Description
[31:23]	Reserved	Reserved. These bits must be set to 0.
[22:6]	Value	Used for locating the start address of a page in flash. Used by the erase and sign commands for specific page addresses.
[5:0]	Reserved	The six reserved bits for byte addresses. The lower six bits of a byte address are ignored because the sign command uses the page address. Returns 0x0 if read.

Flash Controller Upper Page Address Register

Address: 0x4001801C, Reset: 0x00000000, Name: FEEADR1H

Table 55. FEEADR1H Register Bit Descriptions

Bits	Name	Description
[31:23]	Reserved	Reserved. These bits must be set to 0.
[22:6]	Value	Used for locating the end address of a page in flash. Used only by the sign command.
[5:0]	Reserved	The six reserved bits for byte addresses. The lower six bits of a byte address are ignored because the sign command uses the page address. Returns 0x0 if read.

Flash Controller Key Register

Address: 0x40018020, Reset: 0x0000, Name: FEEKEY

Table 56. FEEKEY Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Enter 0xF456 followed by 0xF123. Returns 0x0 if read.

Program Flash Write Protection Register

Address: 0x40018028, Reset: 0xFFFFFFFF, Name: FEEPROP

Table 57. FEEPROP Register Bit Descriptions

Bits	Name	Description
[31:0]	Value	Write protection for program flash. This register is read only if the write protection in flash has been programmed. 0: protect a section of flash. 1: leave a flash block unprotected.

Data Flash Write Protection Register

Address: 0x4001802C, Reset: 0xFF, Name: FEEPROD

Table 58. FEEPROD Register Bit Descriptions

Bits	Name	Description
[7:0]	Value	Write protection for data flash. This register is read only if the write protection in flash has been programmed. 0: protect a section of flash. 1: leave a flash block unprotected.

Data Flash ECC Disable Register

Address: 0x40018030, Reset: 0x00000000, Name: FEEECC

This register is key protected. To write to the register, the user must first write 0xF456 followed by 0xF123 to the key register.

Table 59. FEEECC Register Bit Descriptions

Bits	Name	Description
[31]	ECCDISDT_EN	Data flash page ECC disable. 0: ECC enabled for all data flash pages. 1: ECC disabled for data flash pages set in Bits[2:0].
[30:3]	Reserved	Reserved. These bits must be set to 0.
[2:0]	ECCDISDT	Page number for which ECC is to be disabled in data flash. Only one page can be disabled at a time.

Table 60. FEEECC Register Write Examples

Register FEEECC, Bits[31:0]	Result on Data Flash Page
0x80000000	Page 0 ECC disabled
0x80000001	Page 1 ECC disabled
0x80000002	Page 2 ECC disabled
0x80000003	Page 3 ECC disabled
0x80000004	Page 4 ECC disabled
0x80000005	Page 5 ECC disabled
0x80000006	Page 6 ECC disabled
0x80000007	Page 7 ECC disabled
0x0000000X	ECC enabled on all pages

Flash Controller Signature Register

Address: 0x40018034, Reset: 0x00000000, Name: FEESIGN

Table 61. FEESIGN Register Bit Descriptions

Bits	Name	Description
[31:24]	Reserved	Reserved
[23:0]	Value	Signature, Bits[23:0]

Serial Wire Control Register

Address: 0x40018038, Reset: 0x0001, Name: FEECON1

The FEECON1 register is key protected. To write to the FEECON1 register, the user keys must be entered in the FEEKEY register. After writing to FEECON1, a 16-bit value must be written again to the FEEKEY register to lock in the key protection. When serial wire debug mode is disabled, the only way to access the device is via the USERFAKEY registers. Contact Analog Devices to access these registers.

Table 62. FEECON1 Register Bit Descriptions

Bits	Name	Description
[15:1]	Reserved	Reserved. These bit must be set to 0.
0	DBG	Serial wire debug enable. The kernel sets this bit to 1 when the kernel has finished executing, enabling debug access for the user. 0: disable access via the serial wire debug interface. 1: enable access via the serial wire debug interface.

Flash Controller Write Abort Address Register

Address: 0x40018040, Reset: 0x00000000, Name: FEEABORT

Table 63. FEEABORT Register Bit Descriptions

Bits	Name	Description
[31:0]	Value	If a write is aborted, these bits contain the address of the location being written when the write was aborted. This register has an appropriate value if a command abort occurred. This register is read after the command is aborted and must be read before any other command is given. After a reset, the value is 0x0. However, after the initial signature check is completed, the value can be random.

Flash Controller Abort Enable Register

Address: 0x40018048, Reset: 0x0000, Name: FEEAEN0

Table 64. FEEAEN0 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	To allow a system interrupt to abort a write or a command (erase, sign, or mass verify), write a 1 to the appropriate bit in this register. The appropriate bit is determined by the interrupt required to abort the flash command. For example, if the external IRQ1 is required to abort a flash command, set FEEAEN0 = 0x4. Register FEEAEN0, Bits[13:0] enable Interrupt 0 to Interrupt 13 to abort the flash operation. See Table 4 for more information.

Flash Controller User Failure Analysis Key Register

USERFAKEY Low Register, Bits[31:0]

Address: 0x40018068, Reset: 0x00000000, Name: USERFAKEY0

Table 65. USERFAKEY0 Register Bit Descriptions

Bits	Name	Description
[31:0]	Value	User failure analysis key, Register USERFAKEY, Bits[31:0]. The user FA key is a 64-bit key that is used to disable user read protection. It is the responsibility of the user to program this key to a value. To enable this operation, 0xF456 followed by 0xF123 must first be written to the FEEKEY register to prevent accidental setting before the user FA key can be written. This key must be shared with Analog Devices in case failure analysis is required to enable access to user code.

USERFAKEY High Register, Bits[63:32]

Address: 0x4001806C, Reset: 0x00000000, Name: USERFAKEY1

Table 66. USERFAKEY1 Register Bit Descriptions

Bits	Name	Description
[31:0]	Value	Register USERFAKEY, Bits[63:32]

Flash Controller ECC Registers

Program Flash Address for ECC Error Register

Address: 0x40018074, Reset: 0x00000000, Name: FEEPECC

Table 67. FEEPECC Register Bit Descriptions

Bits	Name	Description
[31:23]	Reserved	Reserved.
[22:3]	Value	Address in program flash where the ECC error is detected. The contents are not cleared on a read.
[2:0]	Reserved	Reserved.

Data Flash Address for ECC Error Register

Address: 0x40018078, Reset: 0x00000000, Name: FEEDECC

Table 68. FEEDECC Register Bit Descriptions

Bits	Name	Description
[31:23]	Reserved	Reserved.
[22:3]	Value	Address in data flash where the ECC error is detected. The contents are not cleared on a read.
[2:0]	Reserved	Reserved.

SRAM

SRAM INTERFACE FEATURES

The ADuCM300 features 6 kB of SRAM organized as 6144 data bytes, that is, 1536 words, which are located at 0x20000000. The SRAM space can be used as data memory and as a volatile program space.

SRAM Integrity

The ADuCM300 implements ECC on the SRAM by adding seven bits to 32-bit words. The implemented ECC allows single-bit correction. Any ECC error larger than one bit results in a bus fault exception. One-bit ECC errors can generate an SRAM ECC interrupt by enabling Register SRAMCTRL, Bit 1.

SRAM INITIALIZATION

After a power-on reset or any reset event, the kernel checks the integrity of the SRAM for ECC errors. If a bus fault exception is triggered during the SRAM integrity check in the kernel, the entire SRAM is reinitialized with valid data (0) and a software reset is generated.

SRAM MEMORY MAPPED REGISTERS

Table 69. SRAM Memory Mapped Registers (Base Address 0x40002000)

Offset	Name	Description	Access	Default
0x002C	SRAMCTRL	SRAM control register	Read/write	0x0001
0X0030	SRAMERR	SRAM error location register	Read	0x0000

SRAM Control Register

Address: 0x4000202C, Reset: 0x0001, Name: SRAMCTRL

Table 70. SRAMCTRL Register Bit Descriptions

Bits	Name	Description
15	SRAM initialization complete	This is a read only bit. This bit is valid only if Bit 2 is set. 0: cleared to 0 on power up. 1: set to 1 after SRAM initialization is complete to indicate that all SRAM locations are written to 0 with valid ECC codes.
[14:3]	Reserved	Reserved. These bits must be set to 0.
2	SRAM initialization enable	0: cleared to 0 on power-up. This bit is self cleared and returns to 0 by itself. The user does not need to clear the bit after setting it. 1: set by the user to enable SRAM initialization.
1	Single bit error interrupt enable	0: cleared by the user to disable interrupt. This bit is cleared by the kernel after power-up. 1: set by the user to enable the interrupt in the event that a single-bit SRAM error is detected and corrected by ECC.
0	Reserved	Reserved. This bit must be set to 1.

SRAM Error Location Register

Address: 0x40002030, Reset: 0x0000, Name: SRAMERR

Table 71. SRAMERR Descriptions

Bits	Name	Description
[15:0]	SRAM error location	This is a read only register. The read only register contains the address of the last SRAM access that caused an ECC error (one bit or two bits). The address is an offset to the start address of 0x20000000.

ADC

ADC FEATURES AND OVERVIEW

The ADuCM300 incorporates two Σ-Δ, analog-to-digital converters (ADCs), ADC0 and ADC1.

ADC0, a 20-bit ADC (19 data bits, 1 sign bit) that accepts a differential input and is ideally suited to interface external sensors with low level signal amplitude outputs. The ADuCM300 also has a low noise PGA for ADC0 and a precision, low drift voltage reference

ADC1, primarily a 20-bit ADC, is connected to a flexible input multiplexer. It can measure external single-ended sensor input voltages, external differential input voltages, the internal (on-chip) temperature sensor, or monitor the supply voltage.

These precision measurement channels integrate attenuator on ADC1, a programmable gain amplifier on ADC0, on-chip buffering, Σ-Δ modulators, and digital filtering for precise measurements. The simplified ADC transfer functions are described as follows:

- ADC0 transfer function, where ADC0DAT is the ADC0 conversion result.

$$V_{IN} = \frac{ADC0DAT \times V_{REF}}{2^{28} - 1} \quad (2)$$

where:
V_{IN} is the input voltage.
V_{REF} is the reference voltage.

- ADC1 transfer function for AIN4, where ADC1DAT is the ADC1 conversion result.

$$V_{IN} = \frac{24 \times \left(ADC1DAT \times V_{REF} \right)}{2^{28} - 1} \quad (3)$$

The ADC1 transfer function for AIN5/GND_SW and AIN6/AIN7, where ADC1DAT is the ADC1 conversion result, is shown in Equation 4.

$$V_{IN} = \frac{ADC1DAT \times V_{REF}}{2^{28} - 1} \quad (4)$$

The same transfer function (4) applies to calculate VIN of the internal temperature sensor.
The following additional equations are needed for a temperature value calculation.

$$Temp_{KELVIN} = \frac{V_{IN}}{274\,\mu V} \quad (5)$$

$$Temp_{CELSIUS} = Temp_{KELVIN} - 273.15 \quad (6)$$

The following four options are available for the ADC reference voltage:

- External reference from the VREF pin (default configuration after reset)
- Internal reference (1.2 V) to AGND
- AVDD18 to AGND for ADC0
- AVDD18 to the GND_SW pin for ADC1. The GND_SW pin needs to be connected to ground via an internal grounding resistor or external resistive circuitry. See the ADC Ground Switch section for details.

The supply voltage is 1.8 V from the LDO regulator. The two ADCs can independently select a different reference or use the same reference.

ADC0

The ADC0 allows precision sensing of small or medium signals. As shown in Figure 9, the ADC0 employs a Σ-Δ conversion technique to realize 19 bits plus a sign bit of no missing codes performance..

As shown in Figure 9, the ADC0 employs a Σ-Δ conversion technique to achieve 19 bits plus a sign bit.

Five pairs of differential input signals can be selected inside the input multiplexer as follows:

- AIN0/AIN1
- AIN1/AIN1 (internal short configuration)
- (AVDD18/136)/GND (voltage input to diagnostic)
- AIN2/AIN3

The Σ-Δ modulator converts the sampled input signal into a digital pulse train. The duty cycle of the pulse train contains the digital information relating to the output voltage level. A sinc3 or sinc4 low-pass filter with programmable decimation is applied to the modulator output data stream to produce a valid 20-bit data conversion result at programmable conversion rates ranging from 4 Hz to 8 kHz in normal power mode and 1 Hz to 656 Hz in low power mode.

ADC0 contains logic that allows an interrupt to be generated after a predefined number of conversions or after the resultant conversion has exceeded a programmable threshold value. Additional logic enables a 32-bit accumulator to sum automatically the 20-bit ADC0 result.

The time to a first valid (fully settled) result on the ADC0 is dependent on digital filter settings. See Table 73 for details.

An interrupt can be generated even on unsettled ADC samples by enabling the ADC continuous interrupt option. It is recommended to use interrupts only on fully settled results.

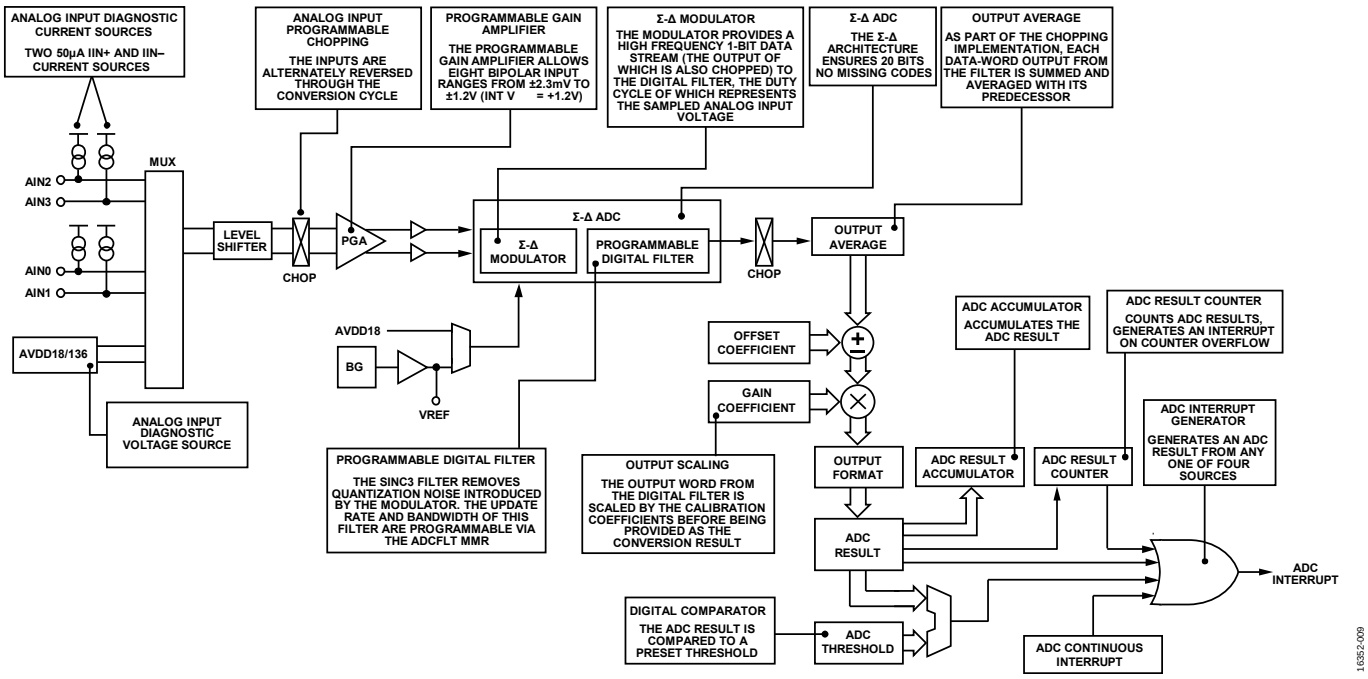

Figure 9. ADC0 Top Level Overview

ADC1

The ADC1 converts additional parameters, for example, voltage and temperature. The input to ADC1 can be multiplexed from an external voltages and an on-chip temperature sensor.

Seven different input signal pairs are selected inside the input mux, as follows:

- AIN4/AGND (input attenuator selected)
- AIN5/GND_SW (external sensor)
- V_{BE1}/V_{BE2} (internal temperature sensor)
- AIN6/AIN7(auxiliary voltage input)
- V_{BE} /GND (VIN for ADC diagnostic)

As shown in Figure 10, the ADC1 employs an identical Σ-Δ conversion technique, including a modified sinc3 or sinc4 low-pass filter to produce a 20-bit data conversion result at programmable output rates ranging from 4 Hz to 8 kHz in normal power mode and 1 Hz to 656 Hz in low power mode.

The AIN4 input is routed to the ADC input via an on-chip, high voltage, divide by 24 resistive attenuator. In this configuration, an external RC filter network is not required.

The device die temperature can be measured using the on-chip temperature sensor. After power-up, the die temperature reflects the environment temperature because self heating has little or no contribution to the internal temperature sensor measurement.

By default, the time to a first valid (fully settled) data conversion result after configuring the input channel is three ADC conversion cycles, with chop mode turned off.

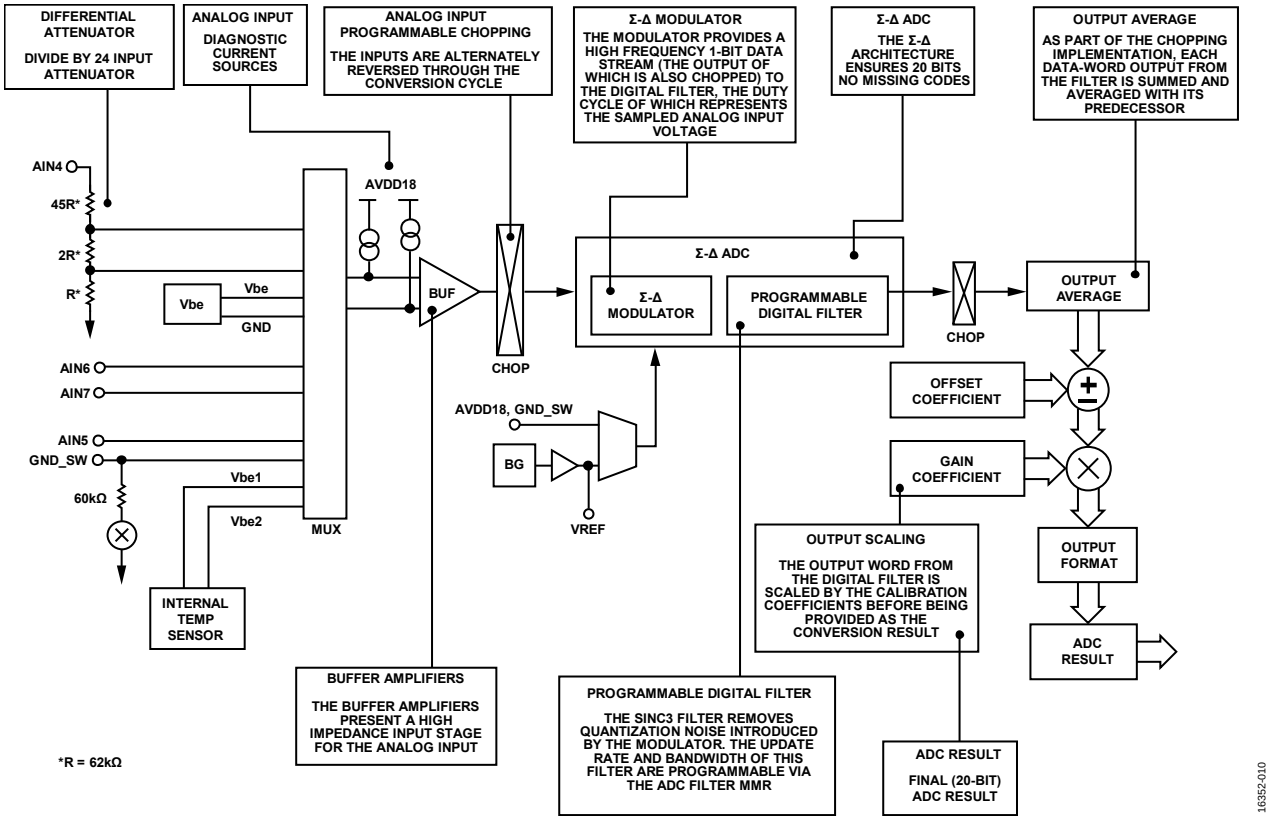

Figure 10. ADC1Top Level Overview

ADC OPERATION

Power Modes

The two power modes are available on the ADuCM300: normal power mode and low power mode.

The ADCs can be configured for reduced (low power) or full power (normal) mode of operation using Register ADCMDE, Bit 3. The Cortex-M3 processor can also be configured to operate in low power mode by configuring the PWRMOD register, Bits[2:0]. The processor power modes are independently controlled and are not related to the ADC power modes described in the ADC Normal Power Mode section and the ADC Low Power Mode section.

Refer to the ADuCM300 data sheet for details on current consumption.

ADC Normal Power Mode

In ADC normal power mode, ADC0 and ADC1 are fully enabled, and the ADC modulator clocked with 512 kHz. The user defined default ADC update rate for all channels in this mode is 1 kHz with ADC chop enabled unless noted otherwise.

ADC Low Power Mode

In ADC low power mode, only ADC0 is enabled in a reduced power and accuracy configuration. The ADC modulator is clocked with 128 kHz. All of the ADC peripheral functions (result counter, digital comparator, and accumulator) can be enabled in low power mode, requiring no extra power. The ADC low power mode is designed for ADC0 gains of 64 to 512.

Typically, in low power mode, ADC0 is configured to run only at a low update rate and to continuously monitor the input voltage. The processor is in power-down mode and wakes up when ADC0 interrupts the core. This interrupt occurs when ADC0 detects a conversion beyond a preprogrammed threshold, a set point, or a set number of conversions.

Bipolar and Unipolar Configuration

The analog inputs to the ADuCM300 can accept either unipolar or bipolar input voltage ranges.

A bipolar voltage does not imply that the device can handle negative voltages with respect to ground. The input range can vary above or below the common-mode voltage by the value of V_{REF} as long as the absolute input voltage range is not exceeded.

Refer to the ADuCM300 data sheet for ADC absolute input range.

Typical ADC Modes of Operation

The ADC can be configured to operate in one of four different general modes of operation, as follows:

- ADC conversion: can be continuous conversions at a fixed rate or single conversions triggered by software.
- ADC idle mode: the ADC is fully powered on, but held in reset.
- ADC calibration modes: these modes remove any ADC and system errors where possible.
- ADC power-down modes: the ADC is powered down to reduce overall system power consumption.

Conversion Modes

In normal ADC operation mode, the following two conversion modes are possible:

- Single conversion: ADCMDE = 0x2. A single ADC conversion can be initiated in software by setting Bit 1 of the ADCMDE register. After a single conversion is completed, the ADC returns to idle mode.
- Continuous conversion: ADCMDE = 0x1. Continuous conversion mode results in the ADCxDAT register being updated at the sampling rate selected by the ADCFLT register.

When a conversion is complete in either mode, the ready flag in the ADCSTA register is asserted, indicating that the ADC result is present in the ADC0DAT register or the ADC1DAT register for reading. These ready flags can be configured to flag an interrupt flag to the Cortex-M3 processor. If an error occurs in the conversion due to an underrange or overrange error in the input voltage for either ADC, the valid bit in the ADC result is asserted, indicating invalid data and the error bit is set in the appropriate ADCSTA register (Bit 13 or Bit 12).

ADC Idle Mode

In idle mode, the ADC is fully powered on, but held in reset. The ADC enters this mode after calibration or between single conversions. Place the device in idle mode when changing any ADC settings.

ADC Power-Down Mode

In power-down mode, the ADCs and the input buffers are fully powered off for maximum power reduction. Place the device in power-down mode only when the entire device is ready to enter hibernate mode.

Before entering hibernate mode, complete the following ADC power-down sequence:

```
set idle mode             (ADCMDE)
disable 1.2Vref           (HRFCTRL)
disable 1.2Vrefbuffer     (IRFPD)
disable ADC operation     (all other ADC MMRs)
disable ADC interrupts    (ADCMSKI)
disable ADC1              (ADC1CON)
disable ADC0              (ADC0CON)
_DSB()                    (optional)
enter PWDN Mode           (ADCMDE)
```

Modifying ADC Settings

When changing ADC settings, switch the ADCs into idle mode, which holds ADCs in reset while they remain fully settled. The user must avoid powering down the ADCs, either individually via Register ADCxCON, Bit 19 or both at the same time via Register ADCMDE, Bits[2:0].

With data output rates larger than 1 kHz, powering down the ADCs can result in the ADCs not being fully settled when the next conversion begins.

```
set Idle Mode             (ADCMDE)
change ADC1               (ADC1CON)
change ADC0               (ADC0CON)
change ADC operation      (all other ADC MMRs)
Wait(100us)               (optional, but good practice)
set ADC operation Mode    (ADCMDE)
```

ADC Power-Up Sequence

The following ADC power-up sequence must be used after the device exits hibernate mode or the ADC exits power-down mode.

Using the power-up sequence after the device exits hibernate mode or after the ADC exits power-down mode ensures the ADCs are operational and synchronized with the first conversion result.

```
config 1.2Vref            (HRFCTRL)
enable 1.2Vrefbuffer      (IRFPD)
enable ADC1               (ADC1CON)
enable ADC0               (ADC0CON)
set Idle Mode             (ADCMDE)
setup ADC operation       (all other ADC MMRs)
Wait(500us)               (see note below)
set ADC operation Mode    (ADCMDE)
```

Note that for data rates of 4 kHz and 8 kHz with PGA gain = 32 or greater, allow up to 10 ms settling time after ADC0 wake-up from power-down mode.

ADC CALIBRATION

As the top level diagrams of Figure 9 and Figure 10 show, the normal signal path through the ADC channels can be described in the following steps:

1. An input voltage is applied through an input buffer (and PGA, in the case of the ADC0) to the Σ-Δ modulator.
2. The modulator output is applied to a programmable digital decimation filter.
3. The filter output result is averaged, if chopping is used.
4. An offset value (ADCxOF) is subtracted from the result.
5. This result is scaled by a gain value (ADCxGN).
6. The result is formatted as twos complement/unipolar or clamped to positive full scale or negative full scale.

Each ADC input channel selection specific offset and gain correction or calibration coefficients are associated with it. These coefficients are stored in an MMR-based offset register and gain register (ADCxOF and ADCxGN). The offset register and gain register can be used to remove internal ADC related and external system level offset and gain errors.

If the chop bit (Register ADCFLT, Bit 15) is set, internal ADC offset errors are minimized, and an offset calibration may not be required. If chopping is disabled, however, an initial offset calibration is required and may need to be repeated, particularly after a large change in temperature. For further information on chopping, refer to the AN-609 Application Note.

The ADCxOF and ADCxGN registers are loaded at power-on with factory defined calibration values. These calibration values vary from device to device, reflecting the manufacturing variability of internal ADC circuits. These registers can also be overwritten by user code to apply application specific calibration coefficients.

When a system calibration is initiated, the ADC generates its calibration coefficient based on an externally generated zero-scale voltage and full-scale voltage, which are applied to the external ADC input for the duration of the calibration cycle. The coefficients are written in the ADCxDAT MMR of the ADC channels. The coefficients are not automatically written in the ADCxOF MMR or ADCxGN MMR. User code must copy these values to their appropriate registers.

The duration of an offset calibration is a full ADC filter settling time before returning the ADC to idle mode. When a calibration cycle is initiated, any ongoing ADC conversion is immediately halted, the calibration is automatically carried out at an ADC update rate programmed into ADCFLT, and the ADC returns to idle after any calibration cycle. It is strongly recommended that ADC calibration be initiated at as low an ADC update rate as possible (high sinc3 filter decimation factor (SF) value in ADCFLT) to minimize the impact of ADC noise during calibration.

Calibrating ADC1 in Internal Temperature Measurement Mode

No gain and offset calibration is required for internal temperature sensing. By copying the factory calibrated gain and offset coefficients for the internal temperature sensor into the ADC2GN register and the ADC2OF register, the device becomes operational when switching the ADC1 channel to temperature sensing. Refer to the ADC1 Control Register section for details.

The factory calibrated coefficients are stored in the 0x000207EA memory location for the ADC2GN register and 0x000207E8 for the ADC2OF register.

Calibrating ADC1 in AIN4 Measurement Mode

To calibrate the offset and gain of the ADC1 (AIN4), a two-point calibration method should be used. This method consists of converting two known voltages (for example, 8 V and 16 V) to determine the slope and offset of the transfer function. The gain coefficient can be divided by the calculated slope to improve the gain error.

The offset error can be reduced by writing ¾ of the calculated offset (in unipolar codes) into the ADC1OF MMR.

Calibrating ADC0 in AIN0/AIN1 Measurement Mode

A gain calibration, particularly in the context of ADC0 (with internal PGA), may need to be carried out at all relevant system gain ranges, depending on system accuracy requirements.

If it is not possible to apply an external full-scale input signal on all gain ranges, the user can apply a lower input signal and scale the result produced by the calibration. For example, apply a 50% current, divide the ADC0DAT value produced by two, and write this value back into the ADC0GN register.

Because ADC0GN is a 16-bit register, a lower limit must be applied to the input signal for system calibration. The input span (difference between the system zero-scale value and the system full-scale value) must be greater than 40% of the nominal full-scale input range, that is, >40% of V_{REF}/gain.

The on-chip Flash/EE memory can be used to store multiple calibration coefficients. These coefficients can be copied by user code directly into the relevant calibration registers, as appropriate, based on the system configuration.

A factory, or end of line, calibration for ADC0 is a two-step procedure, as follows:

1. Apply the zero-scale input signal or small input signal. Configure the ADC in the required PGA setting and other required settings, and write to Register ADCMDE, Bits[2:0] to perform a system zero-scale calibration. This process writes a new offset calibration value into ADC0DAT. User code must store this value into ADC0OF or into Flash/EE memory.
2. Apply a full-scale input signal for the selected PGA setting. Write to Register ADCMDE, Bits[2:0] to perform a system full-scale calibration. This process writes a new gain calibration value into ADC0DAT. This value must be copied by user software to the ADC0GN MMR or into Flash/EE memory.

To reduce the influence of noise on gain and offset errors during calibration, it is advisable to use either the lowest available conversion rate or to use the average of a sufficient number of samples taken at the desired conversion rate.

The actual gain and the required scaling coefficient for zero gain error, varies slightly from device to device and at different PGA settings. The value downloaded into ADC0GN at power-on reset represents the scaling factor for a PGA gain of 8. There is some level of gain error if this value is used at different PGA settings. User code can run ADC calibrations and overwrite the calibration coefficients to correct the gain error at the current PGA setting.

Understanding the Offset and Gain Calibration

The output of the average block in the ADC signal flow can be considered a fractional number with a span for a positive full-scale or negative full-scale input of approximately ±0.75. The span is less than ±1.0 because there is attenuation in the modulator to accommodate some overrange capacity on the input signal. The exact value of the attenuation varies slightly between devices because of manufacturing tolerances.

The offset coefficient is read from the ADCxOF calibration register. This value is a 24-bit, twos complement number.

A positive value of ADCxOF indicates that, when offset is subtracted from the output of the filter, a negative value is added. The nominal value of this register is 0x0000, indicating zero offset is to be removed. The actual offset of the ADC can vary slightly between devices and at different PGA gains. The offset within the ADC is minimized if chopping mode is active (Register ADCFLT, Bit 15 = 1).

The gain coefficient (ADCxGN) is a unitless scaling factor. The nominal value ($ADCxGN_{NOM}$) of this register equals 0x5555, corresponding to a multiplication factor of 1.3333. This factor scales the nominal ±0.75 signal to produce a full-scale output signal of ±1.0, which is checked for overflow and underflow and converted to twos complement or unipolar mode, as appropriate, before being output to the data register.

The ADC transfer function, taking offset and gain calibration factors into consideration, can be described as follows:

For ADC0 (AIN0/AIN0), with the PGA scale enabled,

$$ADC0DAT = \left(\frac{V_{IN} \times \left(2^{28} - 1\right)}{V_{REF}} \times PGA\ Scale - ADC0OF \times \frac{4}{3} \times \frac{2^9}{2^4} \div Gain \right) \times \frac{ADC0GN}{ADC0GN_{NOM}} \quad (7)$$

For ADC0 (AIN0/AIN1), with the PGA scale disabled,

$$ADC0DAT = \left(\frac{V_{IN} \times \left(2^{28} - 1\right)}{V_{REF}} - ADC0OF \times \frac{4}{3} \times \frac{2^9}{2^4} \div Gain \right) \times \frac{ADC0GN}{ADC0GN_{NOM}} \quad (8)$$

For ADC1 (AIN4),

$$ADC1DAT = \left(\frac{\frac{V_{IN}}{24} \times \left(2^{28} - 1\right)}{V_{REF}} - ADC1OF \times \frac{4}{3} \times \frac{2^9}{2^4} \div Gain \right) \times \frac{ADC1GN}{ADC1GN_{NOM}} \quad (9)$$

For ADC1 (AIN5/GND_SW or AIN6/AIN7),

$$ADC1DAT = \left(\frac{V_{IN} \times \left(2^{28} - 1\right)}{V_{REF}} - ADC2OF \times \frac{4}{3} \times \frac{2^9}{2^4} \div Gain \right) \times \frac{ADC2GN}{ADC2GN_{NOM}} \quad (10)$$

For Equation 7 to Equation 10, note the following:

- $2^9/2^4$ is used to convert the ADCxOF format (24-bit data) to ADCxDAT format (32-bit data).
- 4/3 is used to scale ADCxOF data back to actual data, because ADCxOF is Factor 0.75 of the actual offset.
- $ADCxGN_{NOM}$ = 0x5555.
- ADC works in twos complement mode.

Calibration Mode

There are several calibration modes available on the device, which are controlled by the ADC mode register (see Table 84) and are described in this section. It is strongly recommended that these ADC calibration methods be initiated at the lowest ADC update rate possible (high SF value in ADCFLT) to minimize the impact of ADC noise during calibration. Do not use calibration registers for coarse scaling of input ranges.

Self Offset Calibration: ADCMDE = 0x4

In this mode, an offset calibration of the ADC is performed on any enabled ADC using an internally generated 0 V signal. The calibration is carried out at the user programmed ADC settings as a normal single ADC conversion. The offset calibration result is automatically written to the ADC data register of the respective ADC and represents the offset calibration coefficient required to compensate for the present ADC offset. The user needs to store the results to the ADCxOF register. After a device reset, the ADCxOF register is reloaded with the factory calibration value.

After the self offset calibration is performed, the ADCs enter idle mode.

Self Gain Calibration: ADCMDE = 0x5

In this mode, a gain calibration of the ADC against a selected reference voltage is performed on all enabled ADCs. A gain calibration is a two-stage process and takes twice the time of an offset calibration. The gain calibration result is automatically written to the ADC data register of the respective ADC and represents the gain calibration coefficient required to compensate for the present ADC gain error. The user needs to store the results to the ADCxGN register. The internal full-scale calibration does not work for gain settings greater than 1. After a device reset, the ADC gain register is reloaded with the factory calibration value.

After the self gain calibration is performed, the ADCs enter idle mode.

System Zero-Scale Calibration: ADCMDE = 0x6

In this mode, a zero-scale calibration is performed on enabled ADC channels against an external zero-scale voltage driven at the ADC input pins. Usually, the selected channel is shorted externally. The zero-scale calibration result is automatically written to the ADC data register of the respective ADC and represents the offset calibration coefficient required to compensate for the present input offset seen by the ADC. The user needs to store the results to the ADCxOF register. After a device reset, the ADCxOF register is reloaded with the factory calibration value.

After the self zero-scale calibration is performed, the ADCs enter idle mode.

System Full-Scale Calibration: ADCMDE = 0x7

For the ADC0, a system full-scale calibration can only be used when the PGA prescaling is enabled. For details, refer to the Programmable Gain Amplifier section.

In this mode, a system full-scale calibration is performed using the enabled ADC channels against an external full-scale voltage driven at the ADC input pins. The system full-scale calibration result is automatically written to the ADC data register of the respective ADC and represents the gain calibration coefficient required to compensate for the present input gain error. The user needs to transfer the results to the ADCxGN register. After a device reset, the ADC gain register is reloaded with the factory calibration value.

After the self full-scale calibration is performed, the ADCs enter idle mode.

When the ADC is delivering samples at a set ADC conversion rate frequency (f_{ADC}), the user must take care when modifying the ADCxGN register and ADCxOF register to avoid writing to them while the ADC is reading these registers. Ensure that the ADC is in idle mode when modifying the ADCxGN register and ADCxOF register, or ensure that the ADCxGN register and ADCxOF register are modified between the start of the conversion to $1/f_{ADC}$ − 344 clock cycles (16 MHz).

ADC DIGITAL FILTER RESPONSE

The frequency rcesponse of all ADuCM300 ADCs is dominated by the low-pass filter characteristic of the on-chip sinc3 or sinc4 digital filters. The sinc3 or sinc4 filters are used to decimate the ADC Σ-Δ modulator output data bit stream and to generate a 20-bit data result. The digital filter response is identical for both ADCs and is configured via the 20-bit ADC filter (ADCFLT) register. The filter selection affects the overall throughput rate. Noise resolution of the ADCs is determined by the programmed ADC throughput rate. In the case of the ADC0, the noise resolution is determined by the throughput rate and selected gain.

The overall frequency response and the ADC throughput is dominated by the configuration of the SF bits (Register ADCFLT, Bits[6:0]) and the averaging factor (AF) bits (Register ADCFLT, Bits[13:8]). Due to limitations on the digital filter internal data path, there are some limitations on the allowable combinations of SF and AF that can be used to generate a required ADC output rate. This restriction limits the minimum ADC update in normal power mode to 4 Hz (chop on, AF = 60, SF = 31). The calculation of the ADC throughput rate is detailed in Table 87 for the sinc3 and sinc4 filters.

The default ADCFLT value (0x0007) configures the ADCs for a throughput rate of 1.0 kHz with all other filtering options (chop, running average, averaging factor, and sinc3 modify) disabled. A typical filter response based on this default configuration is shown in Figure 11.

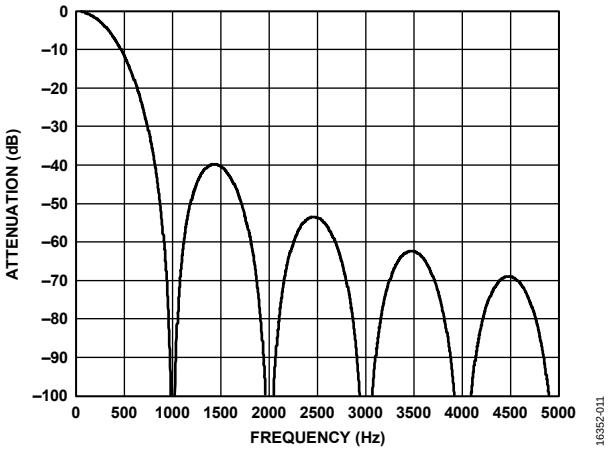

Figure 11. Typical Sinc3 Digital Filter Response at f_{ADC} = 1.0 kHz (ADCFLT = 0x0007)

An additional modify bit, NOTCH2 (Register ADCFLT, Bit 7), is available in the ADCFLT register. This additional notch is set by the user code to modify the standard sinc3 or sinc4 frequency response, increasing the filter stop band rejection by approximately 5 dB. The filter modification is achieved by inserting a second notch (NOTCH2) at

$$f_{NOTCH2} = 1.333 \times f_{NOTCH} \quad (11)$$

where f_{NOTCH} is the location of the first notch in the response. Figure 12 shows the modified 1 kHz filter response when the NOTCH2 bit is active. The new notch is visible at 1.33 kHz. The new notch to an improvement in stop band rejection when compared to the standard 1 kHz response.

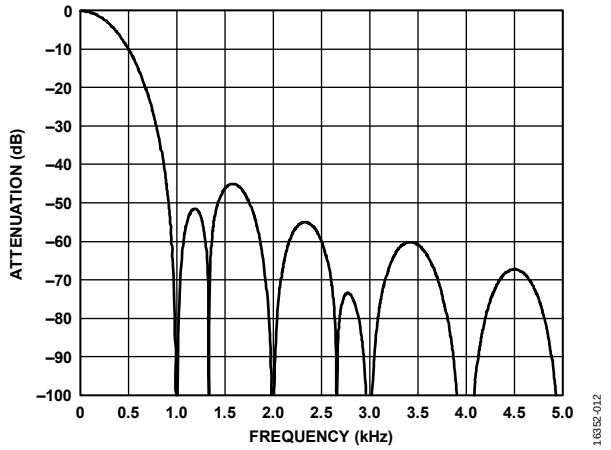

Figure 12. Modified Sinc3 Digital Filter Response at f_{ADC} = 1.0 kHz (ADCFLT = 0x0087)

At very low throughput rates, the chop bit in the ADCFLT register can be enabled to minimize offset errors and, more importantly, temperature drift in the ADC offset error.

Two primary variables (sinc3 decimation factor and averaging factor) are available to allow the user to select an optimum filter response, trading off filter bandwidth against ADC noise. For example, with the chop bit (Register ADCFLT, Bit 15) set to 1, increasing the SF value (Register ADCFLT, Bits[6:0]) to 0x1F (31 decimal) and selecting an AF value (Register ADCFLT, Bits[13:8]) of (22 decimal) 0x16 results in an ADC throughput of 10 Hz. The typical digital filter response is shown in Figure 13.

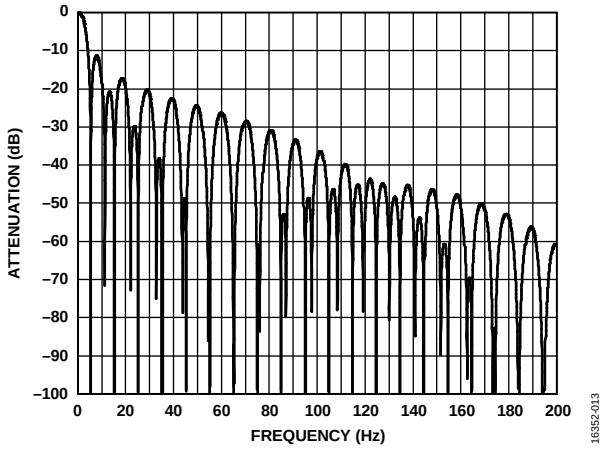

Figure 13. Typical Digital Filter Response at f_{ADC} = 10 Hz (ADCFLT = 0x961F)

In ADC low power mode, the Σ-Δ ADC modulator is clocked four times slower than in normal mode. Subsequently, for the same ADCFLT register configurations, all filter values must be scaled by the same factor.

In general, it is possible to program different values of SF and AF in the ADCFLT register and achieve the same ADC update rate. In practical terms, the trade-off with any value of ADCFLT is frequency response vs. ADC noise. For optimum filter response and ADC noise when using combinations of SF and AF, the best practice is to choose an SF in the range of 16 decimal to 40 decimal, or 0x10 to 0x28, and increasing the AF value to achieve the required ADC throughput. Table 72 shows some common ADCFLT register configurations.

Table 72. Common Sinc3 ADCFLT Register Configurations

ADC Mode	SF	AF	Other Configurations	ADCFLT	f_{ADC}	Settling Time ($t_{SETTLING}$)
Normal	0x1F	0x16	Chop on	0x961F	10 Hz	0.2 sec
Normal	0x07	0x00	Chop off	0x0007	1 kHz	3 ms
Normal	0x07	0x00	Chop off, NOTCH2 on	0x0087	1 kHz	3 ms

DIGITAL FILTER OPTIONS

Example Values SF and AF

The following are examples for AF values and SF values in ADC normal mode:

- f_{ADC} = 1 kHz: chop = 0, AF = 0, SF = 7
- f_{ADC} = 1 kHz: chop = 0, AF = 1, SF = 1 (single cycle settling)
- f_{ADC} = 50 Hz: chop = 0, AF = 0, SF = 127
- f_{ADC} = 10 Hz: chop = 1, AF = 22, SF = 31
- f_{ADC} = 4 Hz: chop = 1, AF = 60, SF = 31 (3.97 Hz)

Enabling the running average by 2 (RAVG2) bit improves noise performance.

An example for an AF value and SF value in ADC low power mode is f_{ADC} = 10 Hz: chop = 1, AF = 2, SF = 39 (modulator clock frequency (f_{MOD}) = 128 kHz).

ADC Conversion Rates and Settling Times

Table 73. ADC Conversion Rates and Settling Times

SINC4_EN	Chop Enabled	Running Average	Averaging Factor	f_{ADC}[1]	$t_{SETTLING}$[2]
No	No	No	No	$f_{MOD} \div (64 \times (SF + 1))$	$3 \div f_{ADC}$
No	No	Yes	No	$f_{MOD} \div (64 \times (SF + 1))$	$4 \div f_{ADC}$
No	No	No	Yes	$f_{MOD} \div (64 \times (SF + 1) \times (3 + AF))$	$1 \div f_{ADC}$
No	No	Yes	Yes	$f_{MOD} \div (64 \times (SF + 1) \times (3 + AF))$	$2 \div f_{ADC}$
No	Yes	Yes or no	Yes or no	$f_{MOD} \div (64 \times (SF + 1) \times (3 + AF) + 3)$	$2 \div f_{ADC}$
Yes	No	No	Not applicable	$f_{MOD} \div (64 \times (SF + 1))$	$4 \div f_{ADC}$
Yes	No	Yes	Not applicable	$f_{MOD} \div (64 \times (SF + 1))$	$5 \div f_{ADC}$
Yes	Yes	Yes or no	Not applicable	$f_{MOD} \div (64 \times (SF + 1) \times 4) + 3)$	$2 \div f_{ADC}$

[1] f_{MOD} = 512 kHz in normal mode and 128 kHz in low power mode.
[2] For $t_{SETTLING}$, an additional 60 µs (approximately) per ADC is required before the first ADC result is available.

Table 74. Allowable Combinations of SF and AF[1]

SF and AF	0	1 to 7	8 to 62
1 to 31	Yes	Yes	Yes
32 to 63	Yes	Yes	No
64 to 127	Yes	No	No

[1] The combination limit of allowed SF settings with AF settings in this table is only for sinc3 mode.

The ADuCM300 also incorporates a sinc4 digital filtering option. The sinc4 filter offers reduced noise, particularly at high (>2 kHz) output rates, and is recommended for this application. Visit the tools and simulations section of the ADuCM300 product page to obtain the ADuCM300 digital filter frequency response calculator showing the ADC frequency response for both sinc3 and sinc4 filtering options.

For sinc4 mode, AF is not applicable, and the user must ensure that AF is always set to zero. SF must be no greater than 0x0F. If SF is set to greater than 0x0F in sinc4 mode, the SF is automatically forced to 0x0F.

FAST TEMPERATURE CONVERSION MODE

The device temperature can be derived through the on-chip temperature sensor. By default, the time to a first valid (fully settled) result after switching the ADC input from the AIN4 to the AIN5/GND_SW channel or from the AIN5/GND_SW to the AIN4 channel is three ADC conversion cycles with chop mode turned off, as shown in Figure 14.

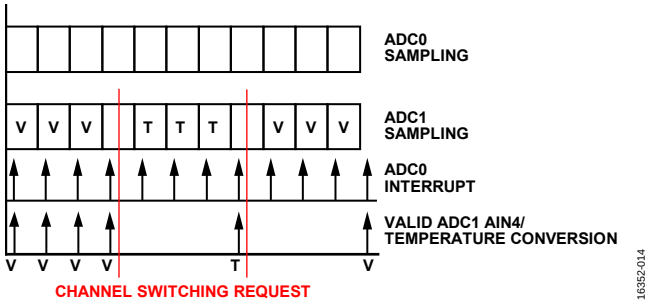

Figure 14. Default Temperature Mode, Chop Off

A fast temperature conversion mode is provided on the internal temperature measurement to minimize the switching delay between voltage conversion (AIN4) and temperature conversions, as shown in Figure 15 and Table 75.

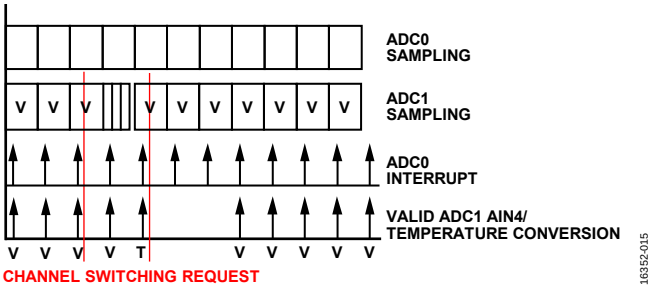

Figure 15. Fast Temperature Mode, Chop Off (ADCFLT = 0x07)

A request for a fast temperature conversion is executed with a delay of one ADC conversion. The fast temperature mode must be cleared after the temperature measurement is available and before a new measurement request.

Table 75. Fast Temperature Mode

ADC0 Interrupts	Valid Flags	User Code
1 to 2	ADC0RDY and ADC1RDY	Voltage = ADC1DAT.
3	ADC0RDY and ADC1RDY	Voltage = ADC1DAT. Set fast temperature request bit.
4	ADC0RDY and ADC1RDY	Voltage = ADC1DAT. This data must be read for the next temperature channel flag to be valid.
5	ADC0RDY and ADC2RDY	Temperature = ADC1DAT. Clear fast temperature request bit.
6 and 7	ADC0RDY	ADC1DAT not valid.
8 to 12	ADC0RDY and ADC1RDY	Voltage = ADC1DAT.

The fast temperature option cannot be used on the first conversion after ADC power-on. The option can only be set after at least the first ADC interrupt. Waiting for a valid ADC result is not necessary. When using the fast temperature mode, ensure that SF ≥ 1. In addition, a conversion rate ≤ 1 ms is recommended in this mode of operation to ensure that the fast result occurs simultaneously with the ADC0 result.

When changing the configuration of the ADCs by writing to the ADCMDE, ADC0CON, or ADCFLT registers, the fast temperature bit in the ADCCFG register must also be cleared to ensure correct operation. This condition is similar to a first conversion after ADC power-on.

ADC DIAGNOSTICS

The ADuCM300 incorporates several hardware features allowing the device to perform on-chip diagnostics, which include the following:

- An AIN0/AIN1 input shorting capability, which is controlled via the ADC0CH bits in the ADC0CON register.
- A defined AVDD18 ÷ 136 input voltage on ADC0, which is controlled via the ADC0CH bits in the ADC0CON register.
- A pair of 50 µA current sources at each ADC channel, which are controlled via the ADCxDIAGx bits in the ADCxCON register.
- A current source for the attenuator in ADC1 in AIN4 configuration, which is controlled via the VE bit in the HVDCFG0 register.

Diagnostic Current Sources

For diagnostic purposes, both ADCs of the ADuCM300 incorporate 50 µA constant current sources. These current sources can be controlled by the appropriate bits in the ADCxCON registers. For all ADC0 input channels (AIN0/AIN1 and AIN2/AIN3), independent current sources are available. Similarly for the ADC1 input channel (AIN5/GND_SW), independent current sources are available. The external restive circuit defines to what extend the diagnostic current sources can be used. The functionality of the diagnostic current sources is explained using ADC0 as an example.

For the implementation of diagnostic functions in the application, the user must take into account the present of the internal resistors R_{AIN0} and R_{AIN1} (see Figure 16). These resistors are untrimmed and have value of 1.7 kΩ with a tolerance of ±25%. Within each device the resistor matching lies within ±120 Ω. The nominal value of the current sources is 50 µA with a tolerance of ±25%. Within each device the current matching lies within ±5 µA. Table 76 shows some possible fault conditions and possible tests for error detection. To conduct these tests, for ADC0 an appropriate gain range must be selected.

Note that for higher gain ranges on ADC0 (gain ≥ 8, ADC0CON[3:0] ≥ 3), in most cases, the diagnostic currents cause an underrange or overrange error status (ADCSTA register, Bit 12 = 1 and the ADC0DAT register, Bit 0 = 1).

Sensing Example

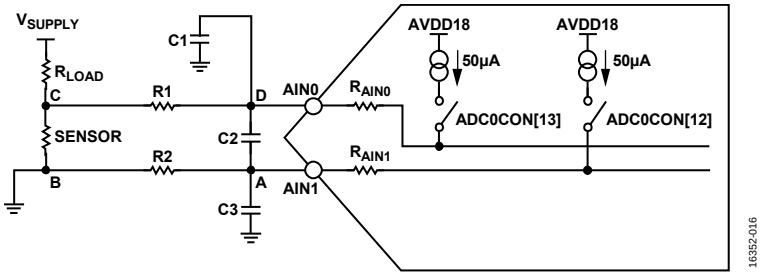

Figure 16. ADC0 Sensing Example Circuit Using Diagnostic Current Sources

Table 76. ADC0 Diagnostics Example

Fault Condition[1]	Tests
Short between Point C and Point D (R1 ~221 Ω shorted) or between Point A and Point D (C2 shorted) or Point D and ground (C1 shorted)	1. Turn off both current sources, and read the value from ADC0DAT. 2. Turn on the current source on AIN0, and read the value from ADC0DAT. 3. If the difference between the two measurements is notably smaller than the nominal value measured in a correct environment, one of the given fault conditions may be present. Note that depending on the actual current flowing through the sensor and the selected gain range, the diagnostic current can cause an overrange error (ADCSTA register, Bit 12 = 1), and the data in ADC0DAT is not valid (ADC0DAT register, Bit 0 = 1).
Short between Point A and Point B (R2 ~221 Ω shorted) or between Point A and Point D (C2 shorted) or Point A grounded (C3 shorted)	1. Turn off both current sources, and read the value from ADC0DAT. 2. Turn on the current source on input AIN1, and read the value from ADC0DAT. 3. If the difference between the two measurements is notably smaller than the nominal value measured in a correct environment, one of the given fault conditions may be present. Note that depending on the actual current flowing through the sensor and the selected gain range the diagnostic current can cause an underrange error (ADCSTA register, Bit 12 = 1), and the data in ADC0DAT is not valid (ADC0DAT register, Bit 0 = 1).
Open circuit between Point C and Point D (R1 unsoldered, or AIN0 unsoldered)	1. Turn on current source on input AIN0, and read the value from ADC0DAT. 2. If the value for gain ranges ≥ 2 (ADC0CON[3:0] ≥ 1) is equal to positive full-scale, the AIN0 input is open circuit. Note that depending on the actual circuitry, calibration, and selected gain range, the diagnostic current can cause an overrange error (ADCSTA register, Bit 12 = 1), and the data in ADC0DAT is not valid (ADC0DAT register, Bit 0 = 1).
Open circuit between Point A and Point B (R2 unsoldered, or AIN1 unsoldered)	1. Turn on the current source on input AIN1, and read the value from ADC0DAT. 2. If the value for gain ranges > 2 (ADC0CON[3:0] ≥ 1) is equal to negative full-scale, the AIN1 input is open circuit. Note that depending on the actual circuitry, calibration, and selected gain range, the diagnostic current can cause an underrange error (ADCSTA register, Bit 12 = 1), and the data in ADC0DAT is not valid (ADC0DAT register, Bit 0 = 1).
High resolution diagnostic allows detection of: (A) R1, R2 mismatch outside of allowed component tolerance (B) A short circuit condition on Point D or Point A, but not both (C) An open circuit condition at R1 (Point D or Point C) or at R2 (Point A or Point B), but not both resistors	1. Turn off both current sources, and read the value from ADC0DAT. 2. Turn on the current source on AIN0 only, and read the value from ADC0DAT. 3. Turn on the current source on AIN1 only, and read the value from ADC0DAT. 4. If for a gain of 1 (ADC0CON[3:0] = 0) the measured changes are not symmetrical, and the differences are not in the range of the nominal measured changes, one of the given fault conditions may be present. Note that depending on the actual circuitry, calibration, and selected gain range, the diagnostic currents can cause an overrange or underrange error (ADCSTA register, Bit 12 = 1), and the data in ADC0DAT is not valid (ADC0DAT register, Bit 0 = 1).

[1] See Figure 16.

For ADC1, configure with AIN4 input channel enabled, an additional constant current sources is available on the high voltage die.

For the AIN4 channel, it is recommended to use the current source as shown in Figure 17. If the current source on the high voltage die in the HVDCFG0 register, Bit 3, is enabled, a typical voltage of approximately 116 mV is added to the actual voltage measured from AIN4, which results in a transferred measurement increase as specified in the data sheet.

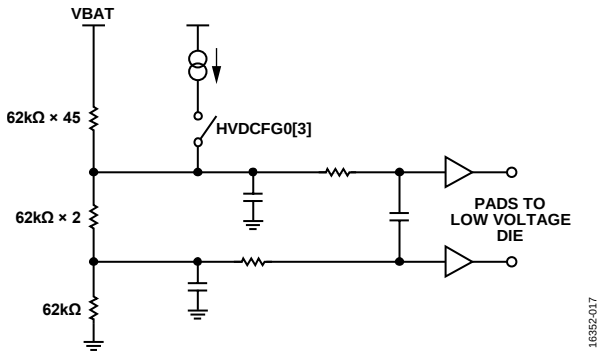

Figure 17. ADC1 Example Circuit Using Diagnostic Current Source on High Voltage Die

OTHER ADC SUPPORT CIRCUITS

Internal and External 1.2 V Voltage Reference

The ADuCM300 has an on-chip 1.2 V high precision voltage reference. It is possible to apply an external reference to the ADuCM300. To use an external reference, the following steps are required:

1. Power down the internal reference buffer by setting IRFPD = 0x01. This setting is also the default state.
2. Apply an external reference to the VREF pin (1.2 V).

The operating mode of the ADCs is not affected by changes to the reference configuration.

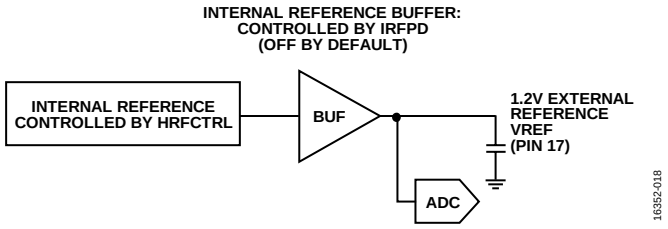

Figure 18. ADC Reference Configuration

Programmable Gain Amplifier

The primary ADC (ADC0) incorporates an on-chip PGA. The gain is controlled by the ADC0CON register. The PGA allows signals of very small amplitude to be amplified while still maintaining low noise performance. Internally, the PGA requires its output range to be limited to <±1 V while still allowing a +1.2 V reference to be used.

To accommodate this requirement, use Bits[11:10] in the ADC0CON register (PGASCALE bits). Clearing these bits (default state) activates a 13/16 × 14/16 scaling.

If the PGASCALE bits are both set to 1, ensure that the input to the PGA does not exceed 1 V/gain.

The PGA is internally divided into two PGA stages with PGA scaling 13/16 applying to stage 1 of the PGA and PGA scaling 14/16 applying to the stage 2 of PGA. The resulting overall PGA scaling factors are listed in Table 77.
Table 77

Table 77. PGA Scaling

Register ADC0CON, Bits[3:0]	PGASCALE	Scaling 14/16	Scaling 13/16	Resulting PGA Scaling Factor Applied
Gain = 2 to 512	11	1	1	Gain × 1
Gain = 2 to 512	10	1	Gain × (13 ÷ 16)	Gain × (13 ÷ 16)
Gain = 32 to 512	01	Gain × (14 ÷ 16)	1	Gain × (14 ÷ 16)
Gain = 32 to 512	00	Gain × (14 ÷ 16)	Gain × (13 ÷ 16)	Gain × (14 ÷ 16) × (13 ÷ 16)

ADC Comparator and Accumulator

ADC0 result can be compared to a preset threshold level (ADC0TH) configured via the ADCCFG register. An interrupt is generated if the absolute (sign independent) value of the ADC result is greater than the preprogrammed comparator threshold level. An ADC interrupt is also generated when the threshold counter reaches a preset value (ADC0RCL).

Finally, a 32-bit accumulator (ADC0ACC) contains the result of multiple primary conversions. User code can read the accumulated value directly (ADC0ACC) without any further software processing.

ADC Ground Switch

The ADuCM300 features an integrated ground switch pin, GND_SW, which allows the user to disconnect ground from external devices and allows a connection to ground using a resistor, reducing the number of external components required for a negative temperature coefficient (NTC) circuit, as shown in Figure 19. The control bit for this switch is in Register ADCCFG, Bit 7. The ground switch feature can be used to reduce power consumption on application specific boards by preventing current flowing through the NTC.

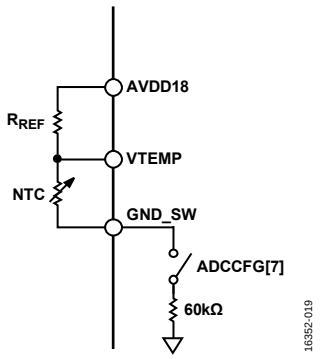

Figure 19. Internal Ground Switch Configuration

ADC CHOPPING

ADC chopping is a technique that is used to cancel the offset and low frequency errors that arise in the signal chain of an ADC. See the AN-609 Application Note for further information. The ADCs on the ADuCM300 implement a chopping scheme where the inputs to the ADC are repeatedly reversed, or chopped. The resulting conversions are inverted from one measurement cycle to the next. By averaging the two input measurements, the offset is removed and the new value updated to the ADC data MMR. The chopping scheme results in low dc offset and offset drift, which is beneficial in applications where drift and noise rejection are required. Chopping is not active by default, meaning that the default configuration has an offset in the conversion result. Two conversion periods are required for an initial measurement result when there is a change to the ADC input, for example, when the channel or gain is changed or following a reset. The effect on settling time is shown in Table 73.

ADC MEMORY MAPPED REGISTERS

Table 78. ADC Reference Control Memory Mapped Registers (Base Address 0x40008800)

Offset	Name	Description	Access	Default
0x0008	HRFCTRL	Internal 1.2 V reference control register	Read/write	0x0002
0x0024	IRFPD	Internal reference buffer power-down register	Read/write	0x0001

Internal 1.2 V Reference Control Register

Address: 0x40008808, Reset: 0x0002, Name: HRFCTRL

Table 79. HRFCTRL Register Bit Descriptions

Bits	Name	Description
[15:2]	Reserved	Reserved. These bits must be set to 0.
1	HRFAUTOB	0: automatic mode. The 1.2 V internal reference is controlled by the ADC. HRFPD is ignored. 1: user mode. The 1.2 V internal reference is controlled by HRFPD. This is the default state.
0	HRFPD	0: enable the 1.2 V internal reference. 1: power down the 1.2 V internal reference.

Internal Reference Buffer Power-Down Register

Address: 0x40008824, Reset: 0x0001, Name: IRFPD

Table 80. IRFPD Register Bit Descriptions

Bits	Name	Description
[15:1]	Reserved	Reserved. These bits must be set to 0.
0	INTREFPD	0: enable internal reference buffer. 1: power down internal reference buffer. Gates in an external 1.2 V reference. This is the default state.

Table 81. ADC Memory Mapped Registers (Base Address 0x40030000)

Offset	Name	Description	Access	Default
0x0000	ADCSTA	ADC status register	Read	0x0000
0x0004	ADCMSKI	ADC interrupt mask register	Read/write	0x00
0x0008	ADCMDE	ADC mode control register	Read/write	0x0003
0x000C	ADC0CON	ADC0 control register	Read/write	0x00000000
0x0010	ADC1CON	ADC1 control register	Read/write	0x00000000
0x0018	ADCFLT	ADC filter configuration register	Read/write	0x00000007
0x001C	ADCCFG	ADC configuration register	Read/write	0x00
0x0020	ADC0DAT	ADC0 result register	Read	0x00000000
0x0024	ADC1DAT	ADC1 result register	Read	0x00000000
0x0030	ADC0OF	ADC0 offset calibration register	Read/write	Calibration value
0x0034	ADC1OF	ADC1 offset calibration register	Read/write	Calibration value
0x0038	ADC2OF	ADC2 offset calibration register	Read/write	Calibration value
0x003C	ADC0GN	ADC0 gain calibration register	Read/write	Calibration value
0x0040	ADC1GN	ADC1 gain calibration register	Read/write	Calibration value
0x0044	ADC2GN	ADC2 gain calibration register	Read/write	Calibration value
0x0048	ADC0RCL	ADC0 result counter limit register	Read/write	0x0001
0x004C	ADC0RCV	ADC0 result counter value register	Read	0x0000
0x0050	ADC0TH	ADC0 comparator threshold register	Read/write	0x00000000
0x0054	ADC0THC	ADC0 threshold counter limit register	Read/write	0x01
0x0058	ADC0THV	ADC0 threshold counter value register	Read	0x00
0x005C	ADC0ACC	ADC0 accumulator register	Read	0x00000000
0x0060	ADC0ATH	ADC0 accumulator threshold register	Read/write	0x00000000

ADC Status Register

Address: 0x40030000, Reset: 0x0000, Name: ADCSTA

All ADC interrupt sources are OR'ed to produce a single ADC interrupt to the Arm core. The user code then reads the ADCSTA MMR to determine the source of the interrupt. The ADC0RDY bit and the ADC1RDY bit are set simultaneously if both the ADC0 and ADC1 are enabled. If the internal temperature sensor or the AIN4 and GND_SW input channel is selected, the ADC2RDY bit is set.

All the RDY bits are cleared by a read of the ADC0DAT MMR. Writing to the ADC0CON register also clears all ADCSTA bits.

A change in the ADC1CON register only clears Register ADCSTA, Bit 1. This bit is also cleared by reading the ADC1DAT register.

If ADC0 is not enabled, all the RDY bits are cleared by a read of the ADC1DAT MMR.

To ensure synchronous sampling, the user reads the ADC1DAT register before reading the ADC0DAT register. Otherwise, a read of the ADC0DAT register clears the RDY1 flag, allowing the possibility of the ADC1DAT register being overwritten by new data before the ADC1DAT register is read.

The following list is a summary of how the status register flags are reset:

- The ADCxERR flags are set or cleared every time new data is written into the appropriate ADCxDAT MMR, setting the ADCxRDY bits. The flags are not cleared by a read of the ADCxDAT MMRs.
- Comparator and counter interrupts are cleared if the comparator or counter is disabled.
- The ADCxRDY bits are automatically cleared by the ADC if the ADC configuration is changed.
- The ADCxRDY flags are cleared by a read of the ADC0DAT register or a read of the ADC1DAT register if the ADC0 is not active.

All other interrupts and flags are also cleared if the ADC configuration is changed.

Table 82. ADCSTA Register Bit Descriptions

Bits	Name	Description
15	ADCxCAL	ADC calibration status 0: cleared to 0 after a write to any of the ADCMDE, ADCFLT, or ADC0CON registers. 1: set to 1 in hardware to indicate an ADC calibration cycle is complete.
14	Reserved	Reserved.

Bits	Name	Description
13	ADC1ERR	ADC conversion error status bit. Error in ADC1 conversion. 0: cleared after reading the ADCSTA register. 1: set to 1 when an underrange or an overrange error occurs in ADC1. This bit is set to 1 when the input voltage at AIN4 exceeds 28.8 V. However, for voltages greater than 29.5 V, this bit is cleared again.
12	ADC0ERR	ADC conversion error status bit. Error in the ADC0 conversion. 0: clear after reading the ADCSTA register. 1: set to 1 when an underrange or an overrange error occurs in ADC0.
[11:7]	Reserved	Reserved.
6	ADCINT	ADC interrupt. 0: cleared to 0 by reading ADC0DAT. 1: if the always interrupt mode is active, this bit generates an interrupt every $1/f_{ADC}$, regardless of whether the digital filter is settled. The ADCxRDY flags must be read to determine if the ADC result is valid. For example, at the default ADCFLT value of 0x0007, this bit is asserted every 1.0 ms. The first two interrupts after reconfiguring the ADC cannot have the ADCxRDY bits set, but all subsequent interrupts can be set. However, using ADC interrupts as a constant 1 ms time base does not guarantee a periodic time base because the ADC time base is interrupted any time the ADC0 is reconfigured.
5	ADC0ATHEX	ADC0 accumulator comparator threshold exceeded. 0: cleared by a reconfiguration of the ADC, or the accumulator comparator is disabled. 1: set if the absolute value of the accumulator exceeds the value written in the ADC0ATH register.
4	ADC0THEX	ADC0 comparator threshold exceeded. Valid only if the ADC0 channel comparator is enabled. 0: cleared by a reconfiguration of the ADC, or if the ADC0TH comparator is disabled. 1: set if the absolute value of the ADC conversion result exceeds the value written in the ADC0TH register. If ADC0THC (ADC threshold counter) is also used, this bit is only set when the specified number of ADC conversions exceed the threshold.
3	ADC0OVR	ADC0 fast overrange bit. 0: change the gain setting to clear this bit. 1: set if the ADC0 input is excessively (>30% approximate on some PGA settings) overrange, and the overrange detect function is enabled. This bit is updated every 125 µs.
2	ADC2RDY	ADC1 (AIN5/GND_SW or internal temperature) conversion result ready bit. 0: cleared to 0 by reading the ADC1DAT and ADC0DAT registers (the registers must be read in this order). Reconfiguring ADC1 or ADC0 also clears this bit. 1: set to 1 at the end of a conversion if ADC1 is enabled and AIN5/GND_SW or internal temperature sensor is selected. This generates an interrupt if the ADC interrupt is enabled and the corresponding bit in the ADCMSKI register is set. If this bit is set, the ADC cannot write further data to the ADC1DAT register. If the always interrupt mode is selected, this bit is a valid flag that indicates when the ADC1 digital filter has fully settled.
1	ADC1RDY	ADC1 (AIN4 or AIN6/AIN7) conversion result ready bit. 0: cleared to 0 by reading the ADC1DAT and ADC0DAT registers (the registers must be read in this order). Reconfiguring the ADC1 or ADC0 also clears this bit. 1: set to 1 at the end of a conversion if ADC1 is enabled and AIN4 is selected. This generates an interrupt if the ADC interrupt is enabled and the corresponding bit in the ADCMSKI register is set. If this bit is set, the ADC cannot write further data to the ADC1DAT register. If the always interrupt mode is selected, this bit is a valid flag that indicates when the ADC1 digital filter has fully settled.
0	ADC0RDY	ADC0 conversion result ready bit. 0: cleared to 0 by reading the ADC0DAT register. Reconfiguring ADC0 also clears this bit. 1: set to 1 at the end of a conversion if ADC0 is enabled. This generates an interrupt if the ADC interrupt is enabled and the corresponding bit in the ADCMSKI register is set. Reading the ADC0DAT register clears this bit. Reconfiguring the ADC0 also clears this bit. If this bit is set, the ADC cannot write further data to the ADC0DAT register. If the ADC result counter is active, RDY is only asserted after the required number of conversions has elapsed. If the always interrupt mode is selected, this bit is a valid flag that indicates when the ADC digital filter has fully settled.

ADC Interrupt Mask Register

Address: 0x40030004, Reset: 0x00, Name: ADCMSKI

This MMR allows the ADC interrupt sources in the ADCSTA register to be individually masked. The bit positions are the same as the lower five bits of ADCSTA. If a bit is set to 1, the interrupt is enabled. The default value is 0x00, that is, all ADC interrupts are inactive.

The ADCMSKI register enables the lower bits of the ADCSTA register to generate an interrupt.

Table 83. ADCMSKI Register Bit Descriptions

Bits	Name	Description
7	Reserved	Reserved. This bit must be set to 0.
6	ADCINT	Mask Register ADCSTA, Bit 6. 0: disable the interrupt (default). 1: enable an interrupt when the ADCINT bit in the ADCSTA register is set.
5	ADC0ATHEX	Mask Register ADCSTA, Bit 5. 0: disable the interrupt (default). 1: enable an interrupt when the ADC0 accumulator comparator threshold is exceeded (the ADC0ATHEX bit in the ADCSTA register is set).
4	ADC0THEX	Mask Register ADCSTA, Bit 4. 0: disable the interrupt (default). 1: enable an interrupt when the ADC0 comparator threshold is exceeded (the ADC0THEX bit in the ADCSTA register is set).
3	ADC0OVR	Mask Register ADCSTA, Bit 3. 0: disable the interrupt (default). 1: enable an interrupt when the ADC0OVR bit in the ADCSTA register is set.
2	ADC2RDY	Mask Register ADCSTA, Bit 2. 0: disable the interrupt (default). 1: enable an interrupt when the ADC2RDY bit in the ADCSTA register is set.
1	ADC1RDY	Mask Register ADCSTA, Bit 1. 0: disable the interrupt (default). 1: enable an interrupt when the ADC1RDY bit in the ADCSTA register is set.
0	ADC0RDY	Mask Register ADCSTA, Bit 0. 0: disable the interrupt (default). 1: enable an interrupt when the ADC0RDY bit in the ADCSTA register is set.

ADC Mode Control Register

Address: 0x40030008, Reset: 0x0003, Name: ADCMDE

A write to the ADCMDE register immediately resets each active ADC, including the ADCxRDY bits and other ADCSTA flags. The full digital filter settling time must elapse before the first result with the new MMR programmed configuration is available. The one exception to this condition is that an interrupt can be provided before the filter is settled if the always interrupt mode bit is set.

Table 84. ADCMDE Register Bit Description

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
5	AINT	Always interrupt. 0: disable this function. 1: enables the ADC to always produce an interrupt $1/f_{ADC}$ after starting a new conversion, even if the digital filter is not settled. The ADCxRDY flags must be read to determine the validity of the ADC data registers. Bit 6 of the ADCSTA and ADCMSKI registers must be used in conjunction with this bit.
4	Reserved	Reserved. This bit must be set to 0.
3	ADCLP	ADC power mode. 0: enable ADC normal power mode (f_{MOD} = 512 kHz). 1: enable ADC low power mode (f_{MOD} = 128 kHz).
[2:0]	ADCMD	ADC mode bits. 000: ADC power-down mode. The ADC circuitry is powered off. This powers down the ADC and PGA. 001: continuous convert mode. The enabled ADC(s) continuously produce conversions at f_{ADC}. The ADCxRDY bits must be cleared to enable new data to be written into the ADCxDAT registers.

Bits	Name	Description
		010: single convert mode. This performs a single conversion on the enabled ADC(s). The ADC enters idle mode after the RDY bit is set. 011: idle mode. The ADC is powered up, but held in reset. The ADC enters idle mode after calibration. 100: self offset calibration. Refer to the Self Offset Calibration: ADCMDE = 0x4 section for details. 101: self gain calibration. Refer to the Self Gain Calibration: ADCMDE = 0x5 section for details. 110: system zero-scale calibration (offset). Refer to the System Zero-Scale Calibration: ADCMDE = 0x6 section for details. 111: system full-scale calibration (system). Refer to the System Full-Scale Calibration: ADCMDE = 0x7 section for details.

ADC0 Control Register

Address: 0x4003000C, Reset: 0x00000000, Name: ADC0CON

A write to the ADC0CON register resets the ADC0 and other enabled ADCs. The full digital filter settling time must elapse before a result is available.

If the two ADCs are being reconfigured at the same time, ADC0CON must be written last to ensure that the ADC1 restarts converting at the same time as ADC0. However, if ADC1 is powered down, there is a power-up delay before ADC1 can start converting. This delay can mean that the ADC0 has started converting, and ADC1 cannot start its conversion until ADC0 has provided an output.

Table 85. ADC0CON Register Bit Descriptions

Bits	Name	Description
19	ADC0EN	Enable ADC0. 0: power down ADC0, RDY bit is cleared. 1: enable ADC0.
18	Reserved	Reserved. This bit must be set to 0.
[17:16]	ADC0BUFBP	ADC0 buffer bypass (positive, negative). 00: no buffer is bypassed. 01: negative buffer is bypassed. 10: positive buffer is bypassed. 11: both buffers are bypassed. This option must be used for hibernate mode only.
[15:14]	ADC0DIAG2	Diagnostic current bits. 00: current source off. 01: enable 50 µA on AIN3. 10: enable 50 µA on AIN2. 11: enable 50 µA on AIN2 and AIN3.
[13:12]	ADC0DIAG1	Diagnostic current bits. 00: current source off. 01: enable 50 µA on AIN1. 10: enable 50 µA on AIN0. 11: enable 50 µA on AIN0 and AIN1.
[11:10]	PGASCALE	PGA scaling bits. If enabled, PGA scaling scales the PGA output by the selected scale factor. To obtain the correct ADC result, user software compensates for the scaling factor selected. For example, if 14/16 PGA scale is selected, user software must compensate by multiplying the ADC0DAT result by 16/14. See the ADC formulas in the Understanding the Offset and Gain Calibration section for further details. 00: PGA gain is scaled by ((14/16) × (13/16)). 01: PGA gain is scaled by 14/16. 10: PGA gain is scaled by 13/16. 11: PGA scaling is disabled. No scaling.
9	ADC0CODE	ADC0 output coding. 0: twos complement (bipolar). This bit must be set to 0. 1: unipolar. In this configuration, any negative input signals have a conversion result of zero.
[8:6]	ADC0CH	ADC0 input channel select. 000: AIN0, AIN1. 001: AIN1, AIN1, internal short configuration. 010: AVDD18 ÷ 136, diagnostic input voltage. 011: AIN2, AIN3, auxiliary input option. 100 to 111: reserved.

Bits	Name	Description
5	Reserved	Reserved. This bit must be set to 0.
4	ADC0REF	ADC0 reference selection. 0: internal reference (1.2 V) or external reference (see Figure 18). 1: AVDD18, AGND.
[3:0]	ADC0PGA	ADC0 gain select: 0000: gain = 1 (only use for a self gain calibration). 0010: gain = 4. 0011: gain = 8. 0100: gain = 16. 0101: gain = 32. 0110: gain = 64. 1001: gain = 512. Others: other selections must not be used.

ADC1 Control Register

Address: 0x40030010, Reset: 0x00000000, Name: ADC1CON

A write to the ADC1CON register resets the ADC1. The ADC1 restarts at an appropriate time so that its outputs are synchronous with ADC0, that is, ADC0 is not reset by a change in the ADC1 configuration. The full ADC1 digital filter settling time must elapse before a ADC1 result is available.

Table 86. ADC1CON Register Bit Descriptions

Bits	Name	Description
19	ADC1EN	Enable ADC1. 0: power down ADC1. ADC1RDY and ADC2RDY bits are cleared. 1: enable ADC1.
18	Reserved	Reserved. This bit must be set to 0.
[17:16]	ADC1BUFBP	ADC1 buffer bypass (positive or negative). 00: no buffer is bypassed. 01: negative buffer is bypassed. 10: positive buffer is bypassed. 11: both buffers are bypassed. This option must be used for hibernate mode only.
[15:14]	Reserved	Reserved. These bits must be set to 0.
[13:12]	ADC1DIAG	Diagnostic current bits. 00: current source off. 01: enable 50 µA to negative input, GND_SW or AIN7. 10: enable 50 µA to positive input, AIN5 or AIN6. 11: enable 50 µA to two differential inputs, AIN5/GND_SW or AIN6/AIN7.
[11:10]	Reserved	Reserved. These bits must be set to 0.
9	ADC1CODE	ADC1 output coding. 0: twos complement (bipolar). 1: unipolar. In this configuration, any negative input signals have a conversion result of zero.
[8:6]	ADC1CH	ADC1 input channel select. 000: AIN4, AGND. AIN4 attenuator. 001: AIN5, GND_SW. External sensor. 010: internal temperature sensor. 011: reserved. 100: AIN6, AIN7, external sensor. 101: reserved. 110: V_{BE}, GND. Nonzero input for ADC diagnostic. 111: reserved.
5	Reserved	Reserved. This bit must be set to 0.

Bits	Name	Description
4	ADC1REF	ADC1 reference selection. 0: internal reference (1.2 V). 1: AVDD18, GND_SW pin. Grounding resistor required via Register ADCCFG, Bit 7, or an external circuit.
[3:0]	Reserved	Reserved. These bits must be set to 0.

ADC Filter Configuration Register

Address: 0x40030018, Reset: 0x00000007, Name: ADCFLT

The ADCFLT register controls the output speed of the ADC, which influences the noise of the ADC. A write to ADCFLT resets the ADCs.

See the ADuCM300 Digital Filter Frequency Response Calculator for more information on settling time and ADC output rate calculations, available from the ADuCM300 product page on the Analog Devices website.

Table 87. ADCFLT Register Bit Descriptions

Bits	Name	Description
16	SINC4_EN	Sinc4 filter enable. 0: sinc filter is in sinc3 mode. 1: sinc filter is in sinc4 mode. In this mode, the user must ensure that the AF bits in Register ADCFLT, Bits[13:8] are set to 0. When the output rate is <2 kHz, this bit must not be set. For an output rate ≥2 kHz, setting this bit makes the filter a sinc4 filter. This bit is recommended when the output rate is greater than 2 kHz because the sinc4 filter is superior to the sinc3 in filtering out the quantization noise.
15	Chop	Enables system chopping of the ADCs. 0: disable system chopping. 1: enable system chopping. Enable chopping to provide very low offset errors and drift. The settling time to a change in configuration equals one conversion period. The bit operates on all ADCs.
14	RAVG2	Enables a running average by 2. 0: disable running average function. 1: enable running average function. RAVG2 implements a simple running average by 2 function to reduce the ADC noise. It is automatically active when chopping is enabled and is an optional feature when chopping is inactive. RAVG2 does not reduce the output rate with the chop bit = 0, but does increase the settling time by one conversion.
[13:8]	AF	Averaging factor. Number of averages = AF + 1. AF implements a programmable, first-order, sinc postfilter. This additional averaging factor further reduces the ADC output rate. There are restrictions on the maximum allowable values of SF and AF. Refer to Table 73 for details.
7	NOTCH2	Inserts a notch at $f_{NOTCH2} = 1.2 \times f_{NOTCH}$. NOTCH2 modifies the frequency response of the sinc3/sinc4 filter to improve the stop band rejection. The worst case rejection at $f > f_{ADC}$ improves to −45 dB, compared to −40 dB with the default sinc3 filter, where f is the sampling frequency. There is a slight increase in ADC noise if this is active. The second notch is generally at (4 ÷ 3) × the main sinc notch to generate notches at both 50 Hz and 60 Hz.
[6:0]	SF	Sinc3 filter decimation factor. Default value: 0x7. SF controls the oversampling rate and decimation factor of the sinc3 filter and sinc4 filter. The output rate from the sinc3 filter and sinc4 filter can be calculated with the equations given in Table 73.

ADC Configuration Register

Address: 0x4003001C, Reset: 0x00, Name: ADCCFG

A write to the ADCCFG register does not reset the ADC.

If the overrange or comparator interrupts are set, the interrupt can be disabled by either clearing the interrupt mask in ADCSTA or disabling the comparator or reconfiguring ADC0.

The comparator must be disabled for at least one full conversion period to ensure that the comparator interrupt is reset. Also, the interrupt mask in ADC0THEX (Register ADCMSKI, Bit 4) must be cleared to prevent unwanted interrupts during this period. The result counter does not produce an interrupt of its own, but only gates the ADCxRDY interrupts. If the result counter and the always interrupt mode are both active, the ADC operation is undefined.

Table 88. ADCCFG Register Bit Descriptions

Bits	Name	Description
7	SWEN	Switch enable. 0: switch from GND_SW pin to AGND is open circuit. 1: switch from GND_SW pin to AGND is closed. This switch is not open circuit automatically when the ADC is placed in power-down mode.
6	Reserved	Reserved. This bit must be set to 0.
5	ADC0ACCEN	ADC0 accumulator enable. 0: accumulator not active, resets accumulator to 0. 1: accumulator active. A negative reading decrements the accumulator. The accumulator can overflow if allowed to run for more than 65,535 conversions.
[4:3]	ADC0CMPEN	ADC0 comparator enable. 00: comparator not active. 01: comparator active. Interrupt if |Data| > ADC0TH. The comparator samples at f_{ADC}, that is, at the ADC output speed. 10: comparator + counter. Interrupt if |Data| > ADC0TH for number of conversions stored in ADC0THC. A value of |Data| < threshold value ($Data_{TH}$) resets the ADC0THCNT counter to 0, if the interrupt is low. 11: comparator + counter. Interrupt if |Data| > ADC0TH for number of conversions stored in ADC0THC. A value of |Data| < $Value_{TH}$ decrements the counter with a floor of 0, if the interrupt is low.
2	ADC0OREN	ADC overrange enable 0: disable this function. 1: enables a coarse overrange comparator on ADC0. If the ADC0 input reading is significantly above positive full scale or below negative full scale for the active PGA setting, the overrange flag (ADC0OVR) in Register ADCSTA, Bit 3 is set. The signal must be outside the full-scale range for 125 µs for this to occur. The ADC conversion is not interrupted if an overrange occurs.
1	Fasten	Fast conversion on internal temperature sensor. 0: disable fast conversion mode on the internal temperature sensor. 1: enable fast conversion mode on the internal temperature sensor. The ADC1 does not switch immediately to the internal temperature sensor. Therefore, there is one more conversion on the channel programmed in ADC1CON, then the ADC switches at the optimum time to the internal temperature sensor to minimize latency. Only a single fast conversion is performed, that is, the ADC only reacts to a 0 to 1 transition on this bit, and, when the transition completes, the ADC reverts to the ADC1CON channel. This bit must be cleared when the fast conversion result is read in preparation for the next fast request. The fast conversion occurs at a fixed conversion speed (1 ms), and must only be used with certain combinations of SF and AF so that the fast result occurs simultaneous with the ADC0 result. This bit cannot be set before the ADC conversions start. This bit must be set after at least one ADC interrupt (this can be an unsettled interrupt if this mode is used).
0	ADCRCEN	ADC result counter enable. 0: counter off and reset. ADC interrupts every 1 ÷ f_{ADC}. 1: counter on. ADC interrupts if ADC0RCV = ADC0RCL. Intermediate ADC conversions are lost if the result counter is active unless the accumulator is active.

ADC0 Result Register

Address: 0x40030020, Reset: 0x00000000, Name: ADC0DAT

Figure 20 shows the formatting of the 32-bit ADC0DAT register according to the gain setting selected.

Only the relevant data bits in the ADC0DAT register are set depending on the gain setting. This means that no software adjustment to the ADC0DAT result is required for different gain settings.

The data format of ADC0DAT is twos complement or unipolar, based on the ADC0CODE bits in the ADC0CON register.

The ADC0DAT register output equals a usable signal range from Bit 18 to Bit 3 at a gain of 512 and noise from Bit 2 to Bit 1.

Register ADC0DAT, Bit 0 indicates if valid data is available. If Register ADC0DAT, Bit 0 = 1, the data is invalid. If Register ADC0DAT, Bit 0 = 0, the data is valid.

The ADC cannot write new data into the ADC0DAT register if the ADC0RDY bit is set. This condition does not apply if the core is off. When the core is off, the ADC0DAT register contains the most recent ADC data.

The data registers are written simultaneously if all ADCs are active, and the ADC0RDY bit is set when the ADC0DAT MMR is written.

Changing the configuration of ADC1 in the middle of a conversion resets ADC1 and does not affect ADC0. ADC0 continues to convert and the data is written into the ADC0DAT register.

If the ADC0RDY bit is low, there is no guarantee that the ADC0DAT MMRs are stable if read.

If the ADC0 result counter is active, the data register (ADC0DAT) is not updated until the ADC0RDY bit is set, that is, when the ADC0RCV counter reaches the programmed limit (ADC0RCL). This condition does not apply if the core is off. If the core is off, the ADC0DAT MMR contains the most recent ADC result(s) when the core wakes up.

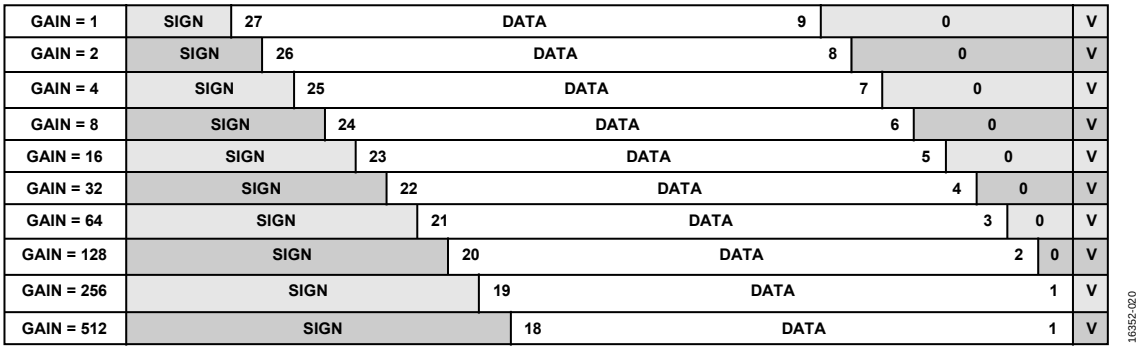

Figure 20. ADC Output (Gain = 1 to Gain = 512)

Table 89. ADC0DAT Register Bit Descriptions

Bits	Name	Description
[31:1]	Result	ADC0 conversion result.
0	Valid	0: data is valid. 1: data is not valid.

ADC1 Result Register

Address: 0x40030024, Reset: 0x00000000, Name: ADC1DAT

The ADC1DAT register has a fixed data width because there is no gain control on ADC1.

The data format of ADC1DAT is twos complement or unipolar, based on the ADC1CODE bits in the ADC1CON registers.

Register ADC1DAT, Bit 0 indicates if valid data is available. If Register ADC1DAT, Bit 0 = 1, the data is invalid. If Register ADC1DAT, Bit 0 = 0, the data is valid.

The ADC cannot write new data into the ADC1DAT register if the relevant ADC1RDY bits are set. This condition does not apply if the core is off. When the core is off, the ADC1DAT register contains the most recent ADC data.

The data registers are written simultaneously if all ADCs are active and the ADCxRDY bits are set when the ADCxDAT MMRs are written.

Changing the configuration of ADC1 in the middle of a conversion resets ADC1. The ADC1DAT register does not update until fully settled output data is available.

If the ADC1RDY bit is low, there is no guarantee that the ADC1DAT MMR is stable if read.

Table 90. ADC1DAT Register Bit Descriptions

Bits	Name	Description
[31:1]	Result	ADC1 conversion result.
0	Valid	0: data is valid. 1: data is not valid.

ADC0 Offset Calibration Register

Address: 0x40030030, Reset: calibration value, Name: ADC0OF

If performing a manual offset calibration with an external shorted input, the Register ADC0DAT, Bits[31:0] result can be transferred to the ADC0OF register using the following formula:

$$ADC0OF = (ADC0DAT \div 2^5) \times PGAGN \times (0x4000 \div ADC0GN) \quad (12)$$

Table 91. ADC0OF Register Bit Descriptions

Bits	Name	Description
[23:0]	ADC0OF	ADC0 24-bit offset calibration coefficient ADC0 measurements. Bit 23 is a sign bit. In offset calibration mode, the result of the calibration is stored in Register ADC0DAT, Bits[23:0]. The user must transfer the result to Register ADC0OF, Bits[23:0].

ADC1 Offset Calibration Register

Address: 0x40030034, Reset: calibration value, Name: ADC1OF

If performing a manual offset calibration, the Register ADC1DAT, Bits[31:0] result can be transferred to the ADC1OF register using the following formula:

$$ADC1OF = (ADC1DAT \div 2^5) \times (0x4000 \div ADC1GN)$$

Table 92. ADC1OF Register Bit Descriptions

Bits	Name	Description
[23:0]	ADC1OF	ADC1 24-bit offset calibration coefficient for AIN4/AGND and AIN6/AIN7 measurements. Bit 23 is a sign bit. In offset calibration mode, the result of the calibration is stored in Register ADC1DAT, Bits[23:0]. The user must transfer the result to Register ADC1OF, Bits[23:0].

ADC2 Offset Calibration Register

Address: 0x40030038, Reset: calibration value, Name: ADC2OF

If performing a manual offset calibration, the Register ADC1DAT, Bits[31:0] result can be transferred to the ADC2OF register using the following formula:

$$ADC2OF = (ADC1DAT \div 2^5) \times (0x4000 \div ADC2GN) \quad (13)$$

Table 93. ADC2OF Register Bit Descriptions

Bits	Name	Description
[23:0]	ADC2OF	ADC2 24-bit offset calibration coefficient for AIN5/GND_SW, internal temperature sensor and V_{BE}/GND measurement. Bit 23 is a sign bit. In offset calibration mode, the result of the calibration is stored in Register ADC1DAT, Bits[23:0]. The user must transfer the result to Register ADC2OF, Bits[23:0].

ADC0 Gain Calibration Register

Address: 0x4003003C, Reset: calibration value, Name: ADC0GN

Table 94. ADC0GN Register Bit Descriptions

Bits	Name	Description
[15:0]	ADC1GN	ADC0 16-bit gain calibration coefficient for all ADC0 measurements. In gain calibration mode, the result of the calibration is stored in Register ADC0DAT, Bits[15:0]. The user must transfer the result to Register ADC0GN, Bits[15:0].

ADC1 Gain Calibration Register

Address: 0x40030040, Reset: calibration value, Name: ADC1GN

Table 95. ADC1GN Register Bit Descriptions

Bits	Name	Description
[15:0]	ADC1GN	ADC1 16-bit gain calibration coefficient for AIN4/AGND and AIN6/AIN7 measurements. In gain calibration mode, the result of the calibration is stored in Register ADC1DAT, Bits[15:0]. The user must transfer the result to Register ADC1GN, Bits[15:0].

ADC2 Gain Calibration Register

Address: 0x40030044, Reset: calibration value, Name: ADC2GN

Table 96. ADC2GN Register Bit Descriptions

Bits	Name	Description
[15:0]	ADC2GN	ADC2 16-bit gain calibration coefficient for AIN5/GND_SW, internal temperature sensor and V_{BE}/GND measurement. In gain calibration mode, the result of the calibration is stored in Register ADC1DAT, Bits[15:0]. The user must transfer the result to Register ADC2GN, Bits[15:0].

ADC0 Result Counter Limit Register

Address: 0x40030048, Reset: 0x0001, Name: ADC0RCL

Table 97. ADC0RCL Register Bit Descriptions

Bits	Name	Description
[15:0]	ADC0RCL	ADC result counter limit register. This register sets the number of conversions required before an ADC interrupt is generated. This feature must be enabled via the ADCCFG MMR (Register ADCCFG, Bit 0). ADC0 comparator and the ADC0 accumulator are always active at f_{ADC}.

ADC0 Result Counter Value Register

Address: 0x4003004C, Reset: 0x0000, Name: ADC0RCV

Table 98. ADC0RCV Register Bit Descriptions

Bits	Name	Description
[15:0]	ADCRCV	ADC result counter value register. This register holds the current number of completed conversions. This value can be used in conjunction with the accumulator to calculate the average of data measured.

ADC0 Comparator Threshold Register

Address: 0x40030050, Reset: 0x00000000, Name: ADC0TH

Table 99. ADC0TH Register Bit Descriptions

Bits	Name	Description
[31:0]	ADC0TH	ADC 32-bit ADC0 comparator threshold register. The absolute value is used by the comparator.

ADC0 Threshold Counter Limit Register

Address 0x40030054, Reset: 0x01, Name: ADC0THC

Table 100. ADC0THC Register Bit Descriptions

Bits	Name	Description
[7:0]	ADC0THC	ADC threshold counter limit. This register determines how many ADC readings above ADC0TH must occur before the ADC0THEX bit is set (Register ADCSTA, Bit 4), causing an interrupt.

ADC0 Threshold Counter Value Register

Address: 0x40030058, Reset: 0x00, Name: ADC0THV

Table 101. ADC0THV Register Bit Descriptions

Bits	Name	Description
[7:0]	ADC0THV	ADC threshold counter. This register contains the current value of the number of times the threshold (ADC0TH) is exceeded.

ADC0 Accumulator Register

Address: 0x4003005C, Reset: 0x00000000, Name: ADC0ACC

The ADC0ACC register is read only and returns the value of the accumulator. The ADC0ACC register updates not more than two ADC clocks earlier than the ADC0DAT register.

There is no warning if the accumulator overflows. The ADC0RCL register can be used to reset the ADC0ACC register after a suitable number of samples. The number of samples is dependent on the gain range selected. For example, at gain = 32, 256 samples can be accumulated.

The accumulator is a signed twos complement or unipolar register, depending on the setting of the ADC0CODE bit.

The accumulator is reset by disabling the accumulator in Register ADCCFG, Bit 5, or by reconfiguring the ADC. If the user is using the ADCCFG register, the enable must be low for at least one full conversion period to ensure that the accumulator is reset. The interrupt mask in Register ADCMSKI, Bit 5 must also be cleared to prevent unwanted interrupts.

Table 102. ADC0ACC Register Bit Descriptions

Bits	Name	Description
[31:0]	ADC0ACC	ADC accumulator. This register holds the accumulated conversion result in the same format as ADC0DAT.

ADC0 Accumulator Threshold Register

Address: 0x40030060, Reset: 0x00000000, Name: ADC0ATH

Table 103. ADC0ATH Register Bit Descriptions

Bits	Name	Description
[31:0]	ADC0ATH	ADC0 accumulator threshold. The result is in the same format as ADC0DAT.

TIMERS

The ADuCM300 has three timers, as follows:

- Timer 0, general-purpose timer
- Timer 2, wake-up timer
- Timer 3, watchdog timer

Timer 0 is a general-purpose, 32-bit, up or down counter with a programmable prescaler. Timer 0 is clocked directly by the PCLK.

Timer 2 is a 32-bit, up or down counter with a programmable prescaler. Timer 2 is clocked directly by the internal 32.768 kHz low frequency oscillator, UCLK, PCLK, or an external clock on P0.4 (ECLKIN).

Timer 3 is a 16-bit, down counter with a programmable prescaler. Timer 3 is clocked by the internal 32.768 kHz low frequency oscillator. The watchdog timer (Timer 3) is used to recover from an illegal software state.

TIMER SYNCHRONIZATION

The synchronization block diagram (seen in Figure 21) shows the interface between user timer MMRs and the core timer blocks. User code can access all timer MMRs directly, including TxVAL, TxCON, and TxCLRI. Data must then transfer from these MMRs to the core timers (Timer 0, Timer 2, and Timer 3) within the timer subsystem. These core timers are buffered from the user MMR interface by the synchronization (SYNC) block. The principle of the synchronization block is to provide a method that ensures data and other required control signals can cross asynchronous clock domains correctly. An example of asynchronous clock domains is the Cortex-M3 processor running on a 16 MHz clock, and the wake-up timer (Timer 2) running on the low frequency oscillator of 32.768 kHz.

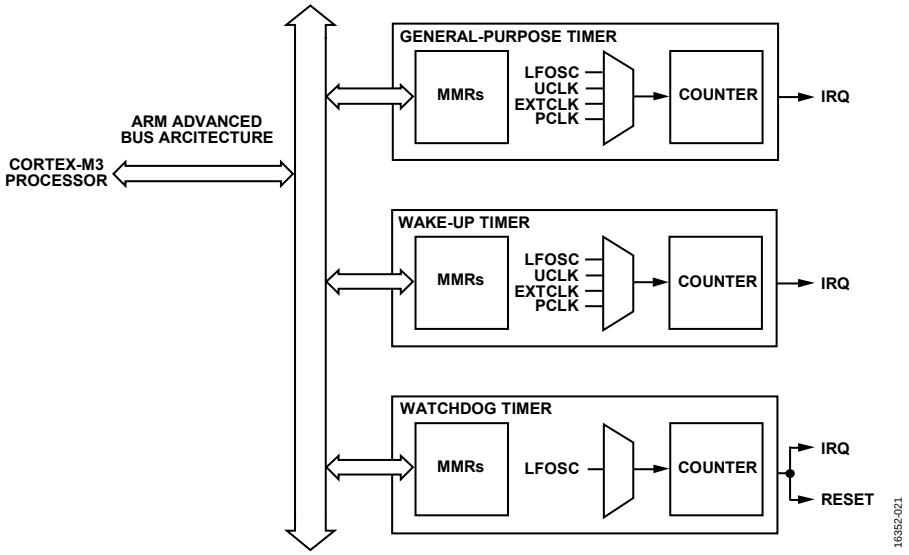

Figure 21. Timer Synchronization Block Diagram

As shown in Figure 21, the MMR logic and core timer logic reside in separate and asynchronous clock domains. Any data coming from the MMR core clock domain and being passed to the internal timer domain must be synchronized to the internal timer clock domain to ensure that it is latched correctly into the core timer clock domain. This synchronization is achieved by using two flip flops, as shown in Figure 22, to not only synchronize, but also to double buffer the data and therefore ensure data integrity in the timer clock domain.

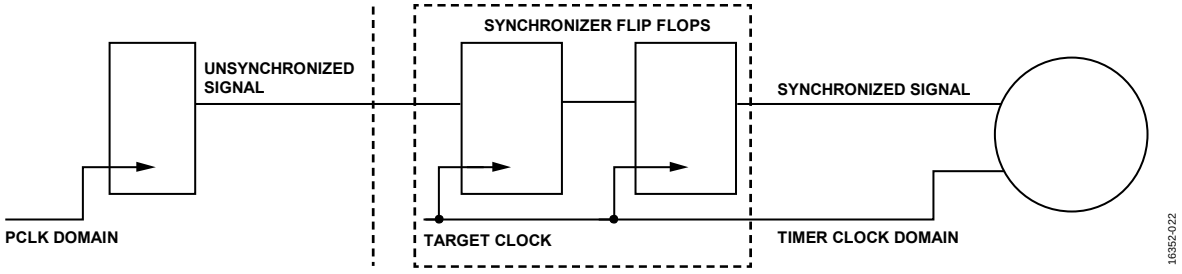

Figure 22. Synchronizer for Signals Crossing Clock Domains

Because of the synchronization block, although timer control data is latched almost immediately (with the fast core clock) in the MMR clock domain, this data in turn cannot reach the core timer logic for at least two periods of the selected internal timer domain clock. It is recommended to wait for three periods of the timer domain clock for critical operations, such as putting the core to sleep.

GENERAL-PURPOSE TIMER

General-Purpose Timer (Timer 0) Features

The ADuCM300 features one general-purpose timer (Timer 0), a 32-bit, count up or count down timer with a programmable prescaler and three timeout fields (A, B, and C), as shown in Figure 23.

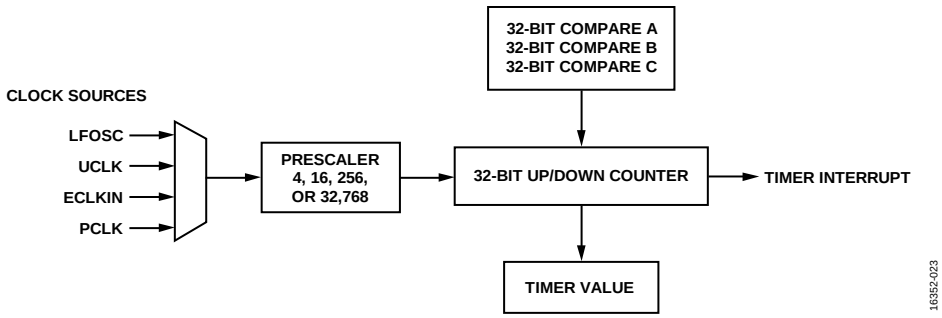

Figure 23. General-Purpose Timers Block Diagram

General-Purpose Timer Overview

The general-purpose timer operates in either a free running or periodic mode. In free running mode, the counter decrements or increments from the maximum or minimum value until the counter reaches zero scale or full scale and starts again at the maximum or minimum value. In periodic mode, the counter decrements or increments until the counter matches the programmed timeout time in the Timeout Field C. This value is stored in the T0TOFC0 register and the T0TOFC1 register. After this value is reached, the timer restarts and counts to the value defined in Field C. Two additional timeout fields are available (Field A and Field B). These fields can be used to set additional timeout cases within the timeout period defined by Field C. A general-purpose timer interrupt (see Table 4) can be enabled for the timeout of any or all of the timeout fields. The value of a counter can be read at any time by accessing the value registers of the counter (T0VALx).

The general-purpose timer (Timer 0) is a 32-bit, up or down counter with a maximum value of 0xFFFFFFFF and a programmable prescaler. The prescaler source is selectable, allowing the clock to be divided by a factor of 4, 16, 256, or 32,768. Timer 0 can be clocked directly by the internal 32.768 kHz low frequency oscillator, UCLK, PCLK, or ECLKIN.

General-Purpose Timer Operation

The general-purpose timer is started by writing 1 to the timer enable bit (Register T0EN, Bit 0), and can be stopped at any time by writing a zero to the same bit. The timer is 32 bits and can be read at any time. In incrementing mode, the timer increments from zero until the counter reaches full scale or the counter matches the programmed timeout value set in Timeout Field C. When the timer reaches the timeout value, a general-purpose timer interrupt (see Table 4) is generated and the timer is reset to zero if interrupts are enabled. In decrement mode, the counter starts at full scale and counts down to the value set in Timeout Feld C, and if configured, generates an interrupt on timeout.

Timer 0 has 32 bits but resides on a 16-bit bus. Therefore, two bus reads are required to obtain the full 32 bits. There are separate addresses for the upper (T0VAL1) 16 bits and lower (T0VAL0) 16 bits of Timer 0. When the freeze control bit (Register T0CON, Bit 3) is enabled, the upper 16 bits are latched and held in a separate register after the lower 16 bits are read. Holding the upper 16 bits in a separate register after the lower 16 bits are read ensures that the counter value can be read in full at a specific time. Both T0VALx registers (upper and lower) remain frozen until the upper 16 bits are read.

Any timer interrupt can be cleared by writing a 1 to the correct clear bit in the T0CLRI register. In the free running mode only, if the counter rolls over, a general-purpose timer interrupt is generated, if enabled.

General-Purpose Timer Configuration

Take care when configuring the general-purpose timer registers to ensure correct operation. The registers can be configured only when the timer is disabled.

On initial startup, all configuration registers must be configured before enabling the timer.

To reconfigure the timer, use the following steps:

1. Set T0EN = 0x0.
2. Wait until Register T0ISTA, Bit 8 = 0. Wait until the enable register is synchronized to the timer clock domain.
3. Write to the register(s) that is or are being reconfigured.
4. Set T0EN = 0x1.

General-Purpose Timer 0 Memory Mapped Registers

Table 104. General-Purpose Timer 0 Memory Mapped Registers (Base Address 0x40000000)

Offset	Name	Description	Access	Default
0x0000	T0VAL0	General-purpose timer current count value (LSB) register	Read	0x0000
0x0004	T0VAL1	General-purpose timer current count value (MSB) register	Read	0x0000
0x0008	T0CON	General-purpose timer control register	Read/write	0x0040
0x000C	T0EN	General-purpose timer enable register	Read/write	0x0000
0x0010	T0TOFA0	General-purpose timer Timeout Field A (LSB) register	Read/write	0x1FFF
0x0014	T0TOFA1	General-purpose timer Timeout Field A (MSB) register	Read/write	0x0000
0x0018	T0TOFB0	General-purpose timer Timeout Field B (LSB) register	Read/write	0x2FFF
0x001C	T0TOFB1	General-purpose timer Timeout Field B (MSB) register	Read/write	0x0000
0x0020	T0TOFC0	General-purpose timer Timeout Field C (LSB) register	Read/write	0x3FFF
0x0024	T0TOFC1	General-purpose timer Timeout Field C (MSB) register	Read/write	0x0000
0x0028	T0IEN	General-purpose timer interrupt enable register	Read/write	0x0000
0x002C	T0ISTA	General-purpose timer interrupt status register	Read	0x0000
0x0030	T0CLRI	General-purpose timer clear interrupts register	Write	0x0000

General-Purpose Timer Current Count Value (LSB) Register

Address: 0x40000000, Reset: 0x0000, Name: T0VAL0

Table 105. T0VAL0 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Current Timer 0 count value, Bit 15 to Bit 0 (LSB).

General-Purpose Timer Current Count Value (MSB) Register

Address: 0x40000004, Reset: 0x0000, Name: T0VAL1

Table 106. T0VAL1 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Current Timer 0 count value, Bit 31 to Bit 16 (MSB).

General-Purpose Timer Control Register

Address: 0x40000008, Reset: 0x0040, Name: T0CON

Table 107. T0CON Register Bit Descriptions

Bits	Name	Description
[15:7]	Reserved	Reserved. These bits must be set to 0.
[6:5]	CLKSEL	Clock select. 00: PCLK. 01: 32.768 kHz low frequency oscillator. 10: UCLK. 11: ECLKIN.

Bits	Name	Description
4	FREE_RUN	Timer free run enable. 0: periodic mode. In increment mode, the timer counts up to T0TOFC0 and T0TOFC1 and returns to 0 and starts again. In decrement mode, the timer counts from 0xFFFFFFFF to T0TOFC0 and T0TOFC1 and starts again at 0xFFFFFFFF. 1: free running mode. In increment mode, the timer counts from 0 to 0xFFFFFFFF and starts again at 0. In decrement mode, the timer counts from 0xFFFFFFFF to 0 and starts again at 0xFFFFFFFF.
3	Freeze	Freeze enable bit. 0: disable this feature (default). 1: enable the freezing of the upper 16 bits (T0VAL1) of the counter after the lower 16 bits are read from T0VAL0. This process of enabling the freezing of the upper 16 bits ensures that the software reads an atomic shot of the timer. T0VAL1 unfreezes after T0VAL1 is read.
2	INC_DEC	Increment or decrement mode. 0: timer starts at zero and starts incrementing. 1: timer starts at full scale and starts decrementing.
[1:0]	PRE	Prescaler. 00: source clock ÷ 4 (default). 01: source clock ÷ 16. 10: source clock ÷ 256. 11: source clock ÷ 32,768.

General-Purpose Timer Enable Register

Address: 0x4000000C, Reset: 0x0000, Name: T0EN

Table 108. T0EN Register Bit Descriptions

Bits	Name	Description
[15:1]	Reserved	Reserved. These bits must be set to 0.
0	T0EN	Timer enable bit. 0: disable the timer. Timer is in a reset state. 1: enable the timer. This bit must be low when configuring the control register or the T0TOFC0 register or T0TOFC1 register in periodic mode. Bit 8 of the status register must also be 0 before programming the control register (or T0TOFC0 or T0TOFC1 in periodic mode).

General-Purpose Timer Timeout Field A (LSB and MSB) Registers

Address: 0x40000010, Reset: 0x1FFF, Name: T0TOFA0

These registers can be written to at any time. However, the corresponding interrupt enable, Bit 0 of the T0IEN register, must be disabled. After these registers are updated, the interrupt can be reenabled.

Table 109. T0TOFA0 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Lower 16 bits of Timeout Field A.

Address: 0x40000014, Reset: 0x0000, Name: T0TOFA1

Table 110. T0TOFA1 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Upper 16 bits of Timeout Field A.

General-Purpose Timer Timeout Field B (LSB and MSB) Registers

Address: 0x40000018, Reset: 0x2FFF, Name: T0TOFB0

These registers can be written to at any time. However, the corresponding interrupt enable, Bit 1 of the T0IEN register, must be disabled. After the register is updated, the interrupt can be reenabled.

Table 111. T0TOFB0 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Lower 16 bits of Timeout Field B.

Address: 0x4000001C, Reset: 0x0000, Name: T0TOFB1

Table 112. T0TOFB1 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Upper 16 bits of Timeout Field B.

General-Purpose Timer Timeout Field C (LSB and MSB) Registers

Address: 0x40000020, Reset: 0x3FFF, Name: T0TOFC0

If the T0VAL register is not free running, it resets after reaching the value of these registers.

For periodic mode, these registers can only be written to when the timer is disabled. In free running mode, these registers can be written to while the timer is running. Before writing to this register, the corresponding interrupt enable, Bit 2 of the T0IEN register, must be disabled. After these registers are updated, the interrupt can be reenabled.

Table 113. T0TOFC0 Register Bit Descriptions

Bits	Name	Description
[1:0]	Value	Lower 16 bits of Timeout Field C

Address: 0x40000024, Reset: 0x0000, Name: T0TOFC1

Table 114. T0TOFC1 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Upper 16 bits of Timeout Field C

General-Purpose Timer Interrupt Enable Register

Address: 0x40000028, Reset: 0x0000, Name: T0IEN

The interrupt enable also acts as an enable for the Timeout Match A, Timeout Match B, and Timeout Match C. If the enable bit is not set, the corresponding status bit is also not set. The corresponding comparison does not occur if the enable bit is not enabled.

Table 115. T0IEN Register Bit Descriptions

Bits	Name	Description
[15:4]	Reserved	Reserved. These bits must be set to 0.
3	Roll	Rollover interrupt enable. Used only in free running mode. 0: disable the rollover interrupt (default). 1: generate an interrupt when the general-purpose timer rolls over.
2	TOFC	T0TOFC interrupt enable. 0: disable T0TOFC interrupt (default). 1: generate an interrupt when T0VAL reaches T0TOFC.
1	TOFB	T0TOFB interrupt enable. 0: disable T0TOFB interrupt (default). 1: generate an interrupt when T0VAL reaches T0TOFB.
0	TOFA	T0TOFA interrupt enable. 0: disable T0TOFA interrupt (default). 1: generate an interrupt when T0VAL reaches T0TOFA.

General-Purpose Timer Interrupt Status Register

Address: 0x4000002C, Reset: 0x0000, Name: T0ISTA

Table 116. T0ISTA Register Bit Descriptions

Bits	Name	Description
[15:9]	Reserved	Reserved.
8	EN_SYNC	Indicates when a change in the enable bit is synchronized to the 32.768 kHz clock domain. 0: cleared automatically when the change in the enable bit has been synchronized to the 32.768 kHz clock domain. 1: set automatically when the enable bit in the T0EN register is set or cleared.
7	Freeze	Timer value freeze. Register T0CON, Bit 3 enables the freeze functionality. 0: cleared automatically when T0VAL1 is not frozen. 1: set automatically to indicate that the value in T0VAL1 is frozen.
[6:4]	Reserved	Reserved.
3	Roll	Rollover interrupt flag. Used only in free running mode. 0: cleared automatically after a write to T0CLRI. 1: set automatically to indicate a rollover interrupt has occurred (the T0VAL counter register is all 1s for increment mode and all 0s for decrement mode) and Register T0IEN, Bit 3 is enabled.
2	TOFC	T0TOFC interrupt flag. 0: cleared automatically after a write to T0CLRI. 1: set automatically to indicate a comparator interrupt has occurred and Register T0IEN, Bit 2 is enabled.
1	TOFB	T0TOFB interrupt flag. 0: cleared automatically after a write to T0CLRI. 1: set automatically to indicate a comparator interrupt has occurred and Register T0IEN, Bit 1 is enabled.
0	TOFA	T0TOFA interrupt flag. 0: cleared automatically after a write to T0CLRI. 1: set automatically to indicate a comparator interrupt has occurred and Register T0IEN, Bit 0 is enabled.

General-Purpose Timer Clear Interrupts Register

Address: 0x40000030, Reset: 0x0000, Name: T0CLRI

Ensure that the register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary.

Table 117. T0CLRI Register Bit Descriptions

Bits	Name	Description
[15:4]	Reserved	Reserved. These bits must be set to 0.
3	Roll	Rollover interrupt clear bit. Used only in free running mode. 0: cleared automatically after synchronization. 1: clear a rollover interrupt flag.
2	TOFC	T0TOFC interrupt flag. 0: cleared automatically after synchronization. 1: clear a T0TOFC interrupt flag.
1	TOFB	T0TOFB interrupt flag. 0: cleared automatically after synchronization. 1: clear a T0TOFB interrupt flag.
0	TOFA	T0TOFA interrupt flag. 0: cleared automatically after synchronization. 1: clear a T0TOFA interrupt flag.

WAKE-UP TIMER

Wake-Up Timer (Timer 2) Features

The wake-up timer (Timer 2) is a 32-bit, up or down counter with a maximum value of 0xFFFFFFFF and a programmable prescaler. The prescaler source is selectable, allowing the clock to be divided by a factor of 1 (or 4, clock source dependent), 16, 256, or 32,768. Timer 2 can be clocked directly by the internal 32.768 kHz low frequency oscillator, UCLK, PCLK, or an ECLKIN.

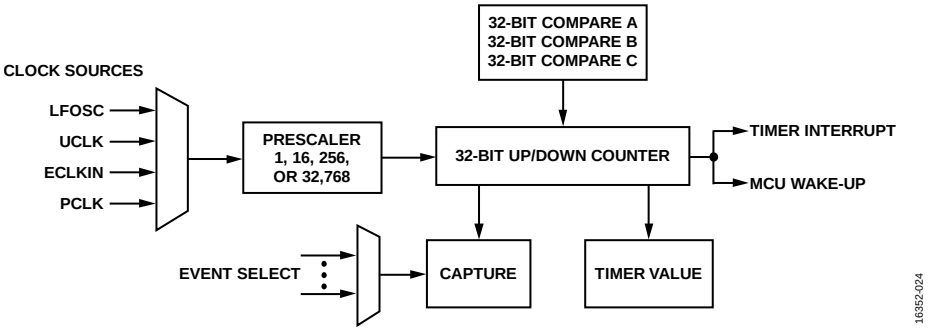

Figure 24. Wake-Up Timer Block Diagram

Wake-Up Timer Overview

Timer 2 can be either free running or periodic. In free running mode, the counter decrements or increments from the maximum or minimum value until the counter is zero scale or full scale, and then starts again at the maximum or minimum value. In periodic mode, the counter decrements or increments from the value in the wake-up field registers (T2WUFCx MMRs) until the counter is zero scale or full scale and starts again at the value stored in the load register. There are three time field values that can be compared with the counter (Field A, Field B, and Field C). The value of a counter can be read at any time by accessing the value register of the counter (T2VALx).

Wake-Up Timer Operation

The wake-up timer (Timer 2) is initiated by setting Bit 0 of the timer enable register to 1 (Register T2EN, Bit 0).

In free running increment mode, the timer increments from zero until the counter reaches full scale. In periodic mode, the counter counts up to the programmed wake-up time in Field C. In decrement mode, the timer starts at full scale and counts down to zero or to the value in Field C. If interrupts are enabled, the counter generates a wake-up timer interrupt (see Table 1) when the counter reaches the target value or the counter rolls over (0xFFFFFFFF to 0x00000000 or vice versa).

Because Timer 2 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 Timer 2. 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. Both T2VALx registers (upper and lower) remain frozen until the upper 16 bits are read. The freeze bit (Register T2CON, Bit 3) must be set to freeze the T2VALx registers between lower and upper reads.

Any timer interrupt can be cleared by writing a 1 to the correct clear bit in the T2CLRI register. In the free running mode only, if the counter rolls over, a wake-up timer interrupt is generated, if enabled.

Timer 2 Configuration

Take care when configuring the wake-up timer registers to ensure correct operation of the timer. The registers can only be configured when the timer is disabled.

On initial startup, all configuration registers must be configured before enabling the timer.

To reconfigure the timer, use the following steps:

1. Set T2EN = 0x0.
2. Wait until Register T2ISTA, Bit 8 = 0. Wait until the enable register is synchronized to the timer clock domain.
3. Write to the register(s) that is or are being reconfigured.
4. Set T2EN = 0x1.

Wake-Up Feature

The wake-up feature is enabled from register programming that selects up to three specific time fields to compare with the corresponding wake-up timer counter. The user is responsible for writing the code to wake up the device. The timer is responsible only for the generation of the wake-up interrupt.

T2WUFA to T2WUFC

After one of these three time fields (T2WUFAx to T2WUFCx) matches the Timer 2 counter, a wake-up event or interrupt can be generated. The top time field (T2WUFC0 and T2WUFC1) value has priority. If the FREE_RUN control bit is cleared, Timer 2 is reset to zero after reaching this wake-up time (or set to full scale if in decrementing mode). The timer continues to increment or decrement, and the wake-up is periodic without software programming. If the FREE_RUN bit is set, wake-up events can be generated, but reprogramming the wake-up time field is required after one or all three time fields are reached (unless full timer length wake-up is desired) because of the timer not being reset after each wake-up event. The interrupt enable bits for these three time values also act as wake-up pulse enable bits. If the time value of a register needs to be changed, after a wake-up occurs, the interrupt enable bit for it must first be disabled because the match logic is in the 32.768 kHz time domain. The disabling of the interrupt bit ensures that another match cannot occur. After the new value is loaded into the time field register, the interrupt can be enabled again.

Wake-Up Timer Memory Mapped Registers

Table 118. Wake-Up Timer Memory Mapped Registers (Base Address 0x40002500)

Offset	Name	Description	Access	Default
0x0000	T2VAL0	Wake-up timer current count value (LSB) register	Read	0x0000
0x0004	T2VAL1	Wake-up timer current count value (MSB) register	Read	0x0000
0x0008	T2CON	Wake-up timer control register	Read/write	0x0010
0x000C	T2EN	Wake-up timer enable register	Read/write	0x0000
0x0010	T2WUFA0	Wake-up timer Wake-Up Field A (LSB) register	Read/write	0x1FFF
0x0014	T2WUFA1	Wake-up timer Wake-Up Field A (MSB) register	Read/write	0x0000
0x0018	T2WUFB0	Wake-up timer Wake-Up Field B (LSB) register	Read/write	0x2FFF
0x001C	T2WUFB1	Wake-up timer Wake-Up Field B (MSB) register	Read/write	0x0000
0x0020	T2WUFC0	Wake-up timer Wake-Up Field C (LSB) register	Read/write	0x3FFF
0x0024	T2WUFC1	Wake-up timer Wake-Up Field C (MSB) register	Read/write	0x0000
0x0028	T2IEN	Wake-up timer interrupt enable register	Read/write	0x0000
0x002C	T2ISTA	Wake-up timer interrupt status register	Read	0x0000
0x0030	T2CLRI	Wake-up timer clear interrupts register	Write	0x0000
0x003C	T2CAP0	Wake-up timer capture event count (LSB) register	Read	0x0000
0x0040	T2CAP1	Wake-up timer capture event count (MSB) register	Read	0x0000

Wake-Up Timer Current Count (LSB and MSB) Value Registers

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

Table 119. T2VAL0 Register Bit Description

Bits	Name	Description
[15:0]	Value	Current Timer 2 count value, Bit 15 to Bit 0 (LSB)

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

Table 120. T2VAL1 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Current Timer 2 count value, Bit 31 to Bit 16 (MSB)

Wake-Up Timer Control Register

Address: 0x40002508, Reset: 0x0010, Name: T2CON

Table 121. T2CON Register Bit Descriptions

Bits	Name	Description
[15:12]	Reserved	Reserved. These bits must be set to 0.
[11:8]	EVENT_SEL	Event select. Select one of the events for event capture. 0000: External Interrupt 0 (P0.3). 0001: External Interrupt 1 (P0.4). 0010: watchdog timer interrupt. 0011: general-purpose timer interrupt. 0100: ADC interrupt. 0101: flash interrupt. 0110: SPI interrupt. 0111: LIN0 interrupt. 1000: LIN1 interrupt. 1001: LIN2 interrupt. 1010: HV interrupt. 1011: Low frequency oscillator calibration interrupt. 1100: SRAM ECC interrupt. 1101: not used. 1110: not used. 1111: not used.
7	EVENT_ENABLE	Event capture enable. 0: disable the event capture logic. 1: enable the event capture logic.
[6:5]	CLKSEL	Clock select. 00: PCLK. 01: 32.768 kHz low frequency oscillator. 10: UCLK. 11: ECLKIN.
4	FREE_RUN	Timer free running enable. 0: periodic mode. In increment mode, the timer counts to T2WUFC0/T2WUFC1 and returns to 0 and starts again. In decrement mode, the timer counts from 0xFFFFFFFF to T2WUFC0/T2WUFC1 and starts again at 0xFFFFFFFF. 1: free running mode. In increment mode, the timer counts from 0 to 0xFFFFFFFF and starts again at 0. In decrement mode, the timer counts from 0xFFFFFFFF to 0 and starts again at 0xFFFFFFFF.
3	Freeze	Freeze enable bit. 0: cleared by user to disable this feature (default).
		1: enable the freezing of the upper 16 bits (T2VAL1) of the counter after the lower bits are read from the T2VAL0 register. This process of enabling the freezing of the upper 16 bits ensures that the software reads an atomic shot of the timer. The T2VAL1 register unfreezes after the T2VAL1 register is read.
2	INC_DEC	Increment or decrement mode. 0: timer starts at zero and starts incrementing. 1: timer starts at full scale and starts decrementing.
[1:0]	PRE	Prescaler. 00: source clock ÷ 1. If the selected clock source is UCLK or PCLK, this setting results in a prescaler of 4. 01: source clock ÷ 16. 10: source clock ÷ 256. 11: source clock ÷ 32,768.

Wake-Up Timer Enable Register

Address: 0x4000250C, Reset: 0x0000, Name: T2EN

Table 122. T2EN Register Bit Description

Bits	Name	Description
[15:1]	Reserved	Reserved. These bits must be set to 0.
0	T2EN	Timer enable bit. 0: disable the timer. Timer is in a reset state. 1: enable the timer. This bit must be low when configuring the control register or the T2WUFC0/T2WUFC1 register in periodic mode. Bit 8 of the status register must also be 0 before programming the control register (or T2WUFC0/T2WUFC1 in periodic mode).

Wake-Up Timer Wake-Up Field A (LSB and MSB) Registers

Address: 0x40002510, Reset: 0x1FFF, Name: T2WUFA0

The T2WUFAx registers can be written to at any time. However, the corresponding interrupt enable, Bit 0 of the T2IEN register, must be disabled. After these registers are updated, the interrupt can be reenabled.

Table 123. T2WUFA0 Register Bit Descriptions

Bits	Name	Description
[15:0]	T2WUFA0	Lower 16 bits of Wake-Up Field A.

Address: 0x40002514, Reset: 0x0000, Name: T2WUFA1

Table 124. T2WUFA1 Register Bit Description

Bits	Name	Description
[15:0]	T2WUFA1	Upper 16 bits of Wake-Up Field A.

Wake-Up Timer Wake-Up Field B (LSB and MSB) Registers

Address: 0x40002518, Reset: 0x2FFF, Name: T2WUFB0

The T2WUFBx registers can be written to at any time, but the corresponding interrupt enable, Bit 1 of T2IEN register, must be disabled. After these registers are updated, the interrupt can be reenabled.

Table 125. T2WUFB0 Register Bit Descriptions

Bits	Name	Description
[15:0]	T2WUFB0	Lower 16 bits of Wake-Up Field B.

Address: 0x4000251C, Reset: 0x0000, Name: T2WUFB1

Table 126. T2WUFB1 Register Bit Descriptions

Bits	Name	Description
[15:0]	T2WUFB1	Upper 16 bits of Wake-Up Field B.

Wake-Up Timer Wake-Up Field C (LSB and MSB) Registers

Address: 0x40002520, Reset: 0x3FFF, Name: T2WUFC0

If the T2VAL register is not free running, the T2VAL register resets after reaching the value of the T2WUFCx registers.

For periodic mode, these registers can only be written to when the timer is disabled. In free running mode, these registers can be written to while the timer is running. Before writing to these registers, the corresponding interrupt enable, Bit 2 of the T2IEN register, must be disabled. After these registers are updated, the interrupt can be reenabled.

Table 127. T2WUFC0 Register Bit Description

Bits	Name	Description
[15:0]	T2WUFC0	Lower 16 bits of Wake-Up Field C.

Address: 0x40002524, Reset: 0x0000, Name: T2WUFC1

Table 128. T2WUFC1 Register Bit Description

Bits	Name	Description
[15:0]	T2WUFC1	Upper 16 bits of Wake-Up Field C.

Wake-Up Timer Interrupt Enable Register

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

The interrupt enable also acts as an enable for the Wake-Up Match A, Wake-Up Match B, and Wake-Up Match C. If the enable bit is not set, the corresponding status bit cannot be set. The corresponding comparison does not occur if the enable bit is not enabled.

Table 129. T2IEN Register Bit Descriptions

Bits	Name	Description
[15:4]	Reserved	Reserved. These bits must be set to 0.
3	Roll	Rollover interrupt enable. Used only in free running mode. 0: disable the roll over interrupt (default). 1: generate an interrupt when the wake-up timer rolls over.
2	WUFC	T2WUFCx interrupt enable. 0: disable T2WUFCx interrupt (default). 1: generate an interrupt when T2VALx reaches T2WUFCx.
1	WUFB	T2WUFBx interrupt enable. 0: disable T2WUFBx interrupt (default). 1: generate an interrupt when T2VALx reaches T2WUFBx.
0	WUFA	T2WUFAx interrupt enable. 0: disable T2WUFAx interrupt (default). 1: generate an interrupt when T2VALx reaches T2WUFAx.

Wake-Up Timer Interrupt Status Register

Address: 0x4000252C, Reset: 0x0000, Name: T2ISTA

Table 130. T2ISTA Register Bit Descriptions

Bits	Name	Description
[15:9]	Reserved	Reserved.
8	EN_SYNC	Indicates when a change in the enable bit is synchronized to the 32.768 kHz clock domain. 0: cleared automatically when the change in the enable bit is synchronized to the 32.768 kHz clock domain. 1: set automatically when the timer enable bit in the T2EN register is set or cleared.
7	Freeze	Timer value freeze. Register T2CON, Bit 3 enables the freeze functionality. 0: cleared automatically when the T2VAL1 register is not frozen. 1: set automatically to indicate that the value in the T2VAL1 register is frozen.
6	Wake-up status	Status of the wake-up signal to the power-down control circuit. These signals are outputs from the timer and wake the core up on a wake-up event and are active high. 0: power-down of the timer can occur immediately. 1: power-down is delayed until all signals are cleared.
5	Reserved	Reserved.
4	Capture	Capture event flag. Indicates when a capture event is pending. 0: cleared automatically when the T2CAP1 register is read. 1: set automatically when a rising edge is detected on the selected event line.
3	Roll	Rollover interrupt flag. Used only in free running mode. 0: cleared automatically after a write to T2CLRI. 1: set automatically to indicate a rollover interrupt has occurred, that is, the T2VALx counter registers are all 1s for increment mode and all 0s for decrement mode.
2	WUFC	T2WUFC interrupt flag. 0: cleared automatically after a write to T2CLRI. 1: set automatically to indicate a comparator interrupt has occurred and Register T2IEN, Bit 2 is enabled.
1	WUFB	T2WUFB interrupt flag. 0: cleared automatically after a write to T2CLRI. 1: set automatically to indicate a comparator interrupt has occurred and Register T2IEN, Bit 1 is enabled.

Bits	Name	Description
0	WUFA	T2WUFA interrupt flag. 0: cleared automatically after a write to T2CLRI. 1: set automatically to indicate a comparator interrupt has occurred and Register T2IEN, Bit 0 is enabled.

Wake-Up Timer Clear Interrupts Register

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

Ensure that the register write has fully completed before returning from the interrupt handler. Use the DSB instruction if necessary.

Table 131. T2CLRI Register Bit Descriptions

Bits	Name	Description
[15:5]	Reserved	Reserved. These bits must be set to 0.
4	CAPTURE_CLR	Interrupt clear bit for the capture event interrupt.
3	Roll	Rollover interrupt clear bit. This bit is used only in free running mode. 0: cleared automatically after synchronization. 1: clear a rollover interrupt flag.
2	WUFC	T2WUFCx interrupt flag. 0: cleared automatically after synchronization. 1: clear a T2WUFCx interrupt flag.
1	WUFB	T2WUFBx interrupt flag. 0: cleared automatically after synchronization. 1: clear a T2WUFBx interrupt flag.
0	WUFA	T2WUFAx interrupt flag. 0: cleared automatically after synchronization. 1: clear a T2WUFAx interrupt flag.

Wake-Up Timer Capture Event Count (LSB and MSB) Registers

Address: 0x4000253C, Reset: 0x0000, Name: T2CAP0

Table 132. T2CAP0 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Lower 16 bits of the count at which the selected event occurred

Address: 0x40002540, Reset: 0x0000, Name: T2CAP1

Table 133. T2CAP1 Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Upper 16 bits of the count at which the selected event occurred

The wake-up event capture logic does not work if the device is put into system halt mode or hibernate mode. The device must be kept in active mode while capturing an event.

WATCHDOG TIMER

Watchdog Timer (Timer 3) Features

The watchdog timer (Timer 3) is a 16-bit, down counter with programmable prescaler.

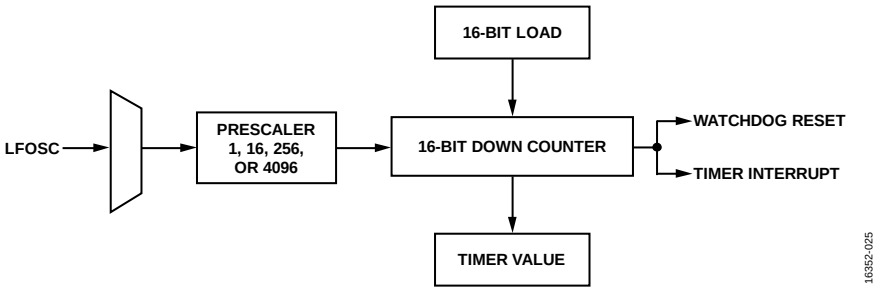

Figure 25. Watchdog Timer Block Diagram

Watchdog Timer Overview

The watchdog timer (Timer 3) is a 16-bit, down counter with a programmable prescaler. The prescaler source is selectable and can be scaled by factors of 1, 16, 256, or 4096.

Timer 3 is used to recover from an illegal software state. After being enabled by the user code, Timer 3 requires periodic servicing to prevent it from forcing a reset or an interrupt of the processor.

Timer 3 is clocked by the internal 32.768 kHz low frequency oscillator. Timer 3 is clocked at all times except during reset.

Timer 3 is active in kernel download mode.

A Timer 3 timeout generates a reset or an IRQ. Register T3CON, Bit 1 is added to allow selection of an IRQ instead of a reset, which can be used for debug purposes. The IRQ can be cleared by writing 0xCCCC to the T3CLRI write only register.

Watchdog Timer Operation

After a POR, the watchdog timer is initialized by the kernel as follows:

T3LD = 0x0400

This initialization enables the watchdog timer with a timeout of 8 sec. This initial configuration can be modified by the following user code:

$$Timeout = PRESCALER \times (T3LD + 1)/32.768 \text{ kHz} \quad (14)$$

However, setting Register T3CON, Bit 5 protects the T3CON register and the T3LD register. After kernel execution, user code can disable the timer once, then reconfigure the timer with Register T3CON, Bit 5 set, only once. After this, only a power cycle can unlock the T3CON and T3LD registers and allow reconfiguration of the timer again. If Register T3CON, Bit 5 is not set by user code, user code can change the T3LD register and the other bits of the T3CON register at any time. If Register T3CON, Bit 5 is cleared, the timer is disabled. The settings can be modified and the timer reenabled.

If the watchdog timer is set to fixed mode (Register T3CON, Bit 6 = 0), the watchdog timer value decrements from 0x1000 to zero, wraps around to 0x1000, and continues to decrement. To achieve a timeout value greater or less than 0x1000 (typically 32 sec with default prescaler = 256), periodic mode must be used (Register T3CON, Bit 6 = 1), and T3LD and Register T3CON, Bits[3:2] (prescaler) be written with the values corresponding to the desired timeout period. The maximum timeout is ~8192 sec (T3LD = 0xFFFF, prescaler = 4096).

When the watchdog timer decrements to zero, a reset (or IRQ) is generated. This reset can be prevented by writing the T3CLRI register with 0xCCCC before the expiration period. A write to the T3CLRI register causes the watchdog timer to reload with T3LD (or 0x1000 if in fixed mode) immediately to begin a new timeout period and start to count again. If any value other than 0xCCCC is written, a reset is generated (or IRQ if selected by Register T3CON, Bit 1).

Watchdog Timer Power-Down Mode

If Register T3CON, Bit 0 is cleared to 0, the watchdog timer continues to count while the Cortex-M3 is halted. If Register T3CON, Bit 0 is set to 1, the watchdog timer is held at its current count while the Cortex-M3 is halted and continues from its previous value after the Cortex-M3 restarts again. This bit must be configured at the same time as Register T3CON, Bit 5.

Watchdog Timer Configuration

Take care when configuring the watchdog timer registers to ensure correct operation of the timer.

Disabling the Watchdog Timer

If Register T3CON, Bit 5 is reset on the initial T3CON write, the timer is reset and disabled.

Reenabling the Watchdog Timer

If Register T3CON, Bit 5 is set in a subsequent T3CON write, the watchdog timer is reset and reloaded or restarted. Depending on the state of Register T3CON, Bit 6, the watchdog timer restarts from 0x1000 (Register T3CON, Bit 6, MOD = 0) or restarts from T3LD (Register T3CON, Bit 6, MOD = 1).

Writing to the T3LD Register

If the T3LD register is written when MOD = 1, the watchdog timer receives the T3LD value and starts counting from there.

Write to T3CLRI to Restart the Watchdog Timer

The user writes 0xCCCC to T3CLRI within the timeout period to restart the watchdog timer normally and prevent a reset or an IRQ. The user has to ensure that at least four low power oscillator clock cycles have lapsed between T3CLR1 writes.

Watchdog Timer Prescaled Decrement

The enabled watchdog timer decrements when the clock divided by prescaler is reached. If the watchdog timer decrements after reaching the minimum value of 0x0000, the watchdog timer is reloaded with either 0x1000 or the T3LD value, depending on the MOD bit. Reaching the minimum value of 0x0000 is a watchdog timeout event. A reset or IRQ is generated, depending on the IRQ bit (Register T3STA, Bit 0).

Watchdog Timer Memory Mapped Registers

Table 134. Watchdog Timer Memory Mapped Registers (Base Address 0x40002580)

Offset	Name	Description	Access	Default
0x0000	T3LD	Watchdog timer load value register	Read/write	0x0400
0x0004	T3VAL	Watchdog timer current count value register	Read	0x1000
0x0008	T3CON	Watchdog timer control register	Read/write	0x00E9
0x000C	T3CLRI	Watchdog timer clear interrupt register	Write	0x0000
0x0018	T3STA	Watchdog timer status register	Read	0x0020

Watchdog Timer Load Value Register

Address: 0x40002580, Reset: 0x0400, Name: T3LD

Table 135. T3LD Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Load value.

Watchdog Timer Current Count Value Register

0x40002584, Reset: 0x1000, Name: T3VAL

Table 136. T3VAL Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Current count value. Read only register.

Watchdog Timer Control Register

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

Table 137. T3CON Register Bit Descriptions

Bits	Name	Description
[15:7]	Reserved	Reserved. These bits must be set to 0.
6	MOD	Timer mode. 0: fixed mode. 1: periodic mode (default). In fixed mode, the timer wraps around at 0x1000.
5	Enable	Timer enable. 0: disable the timer. The timer can only be cleared once. 1: enable the timer (default).
4	Reserved	Reserved. This bit must be set to 0.
[3:2]	PRE	Prescaler. 00: (32,768 ÷ 1) Hz. 01: (32,768 ÷ 16) Hz. 10: (32,768 ÷ 256) Hz (default). 11: (32,768 ÷ 4096) Hz.
1	IRQ	Timer interrupt. 0: generate a reset on a timeout (default). The user sets this bit to 0.
0	PD	Power-down off. 0: enable Timer 3 to continue counting when the Cortex-M3 is halted. 1: halt Timer 3 when the Cortex-M3 is powered down using system halt mode or hibernate mode in the PWRMOD MMR.

Watchdog Timer Clear Interrupt Register

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

Ensure that the register write has fully completed before returning from the interrupt handler. Use the DSB instruction if necessary.

Table 138. T3CLRI Register Bit Descriptions

Bits	Name	Description
[15:0]	Value	Clear watchdog. The user writes 0xCCCC to reset, reload, or restart Timer 3, or clear IRQ. A write of any other value causes a watchdog reset or IRQ. This register is write only and reads back 0.

Watchdog Timer Status Register

Address: 0x40002598, Reset: 0x0020, Name: T3STA

The T3STA register is a read only status register. Because of the asynchronous relationship and frequency difference between the core clock and Timer 3 clock, changes to the watchdog timer configuration are synchronized between the two clock domains. Several of the status bits are used to signal to the core that Timer 3 clock synchronization is in progress. The user can use these bits to verify that the previous timer configuration write has taken effect in the Timer 3 clock domain if necessary.

The status bits typically only need to be used if the watchdog timer is disabled on the initial write to the T3CON register, followed by Timer 3 configuration changes, followed by an immediate reenable of Timer 3. To ensure that all changes are in effect, do not perform the reenable until all in progress status bits are cleared.

Table 139. T3STA Register Bit Descriptions

Bits	Name	Description
[15:5]	Reserved	Reserved.
4	Lock	Lock status bit. 0: cleared by default until user code sets Register T3CON, Bit 5. 1: set automatically in hardware if Register T3CON, Bit 5 has been set by user code.
3	CON_SYNC	T3CON write synchronization. 0: T3CON register write synchronization match. 1: T3CON register write synchronization in progress.
2	LD_SYNC	T3LD write synchronization. 0: T3LD register write synchronization match. 1: T3LD register write synchronization in progress.
1	CLRI_SYNC	T3CLRI write synchronization. 0: T3CLRI write synchronization not complete or inactive. 1: T3CLRI write being synchronized to Timer 3 clock domain. Timer 3 is restarted (if 0xCCCC was written) when synchronization is complete.
0	IRQ	Timer 3 IRQ. 0: Timer 3 interrupt not pending. 1: Timer 3 interrupt pending.

GENERAL-PURPOSE DIGITAL INPUTS AND OUTPUTS

GENERAL-PURPOSE DIGITAL INPUTS AND OUTPUTS OVERVIEW

The ADuCM300 features six bidirectional, GPIO pins. All of the GPIO pins have multiple functions, configurable by user code. These features are described in Table 140.

Each GPIO pin can be configured as an input, output, or open circuit. These pins also have an internal, pull-up, programmable resistor. See the ADuCM300 data sheet for logic input voltages and absolute maximum values. For optimum low power operation, the GPIOs must be configured as inputs (using GP0OEN) and with pull-up resistors enabled (using GP0PUL).

When the ADuCM300 enters a power saving mode, the GPIO pins retain their state.

GENERAL-PURPOSE DIGITAL INPUTS AND OUTPUTS FEATURES

The general-purpose digital inputs and outputs features include the following:

- Six, bidirectional, GPIO pins.
- Internally multiplexed with SPI and LIN blocks.

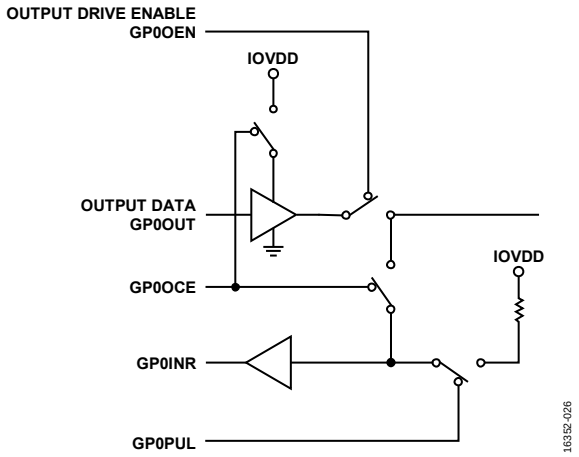

Figure 26. GPIO Block Diagram

GENERAL-PURPOSE DIGITAL PORT MULTIPLEX

This block provides control over the GPIO functionality of specified pins, because some of the pins can work as GPIOs or have other specific functions. The following table details the configuration modes available for each GPIO that can be selected in the GP0CON register.

Table 140. GPIO Multiplex

GPIO	Configuration Mode: 00	Configuration Mode: 01	Configuration Mode: 10	Configuration Mode:11
P0.0	GPIO0	CS	Not applicable	LIN_RX (input)
P0.1	GPIO1	SCLK	Not applicable	LIN_TX (output)
P0.2	GPIO2	MISO	Not applicable	Not applicable
P0.3	GPIO3/IRQ0	MOSI	LC_TX (input)	LIN_TX (output)
P0.4	GPIO4/IRQ1	LC_RX (output)	ECLKIN	LIN_RX (input)
P0.5	GPIO5	LC_TX (output)	Not applicable	LIN_TX (output)

GENERAL-PURPOSE DIGITAL INPUT AND OUTPUT OPERATION

Digital Port Multiplexed Configuration

The pin functions are configured using the GP0CON register. The GP0CON register configures Port 0. External interrupts and input level signals (GP0IN) are available in any of the configuration modes, except when the GPIOs are configured as open circuit or as outputs (GP0OEN = 1 and GP0OCE = 1).

GPIO Pull-Up Enable

All GPIO pins have a switchable internal pull-up resistor. Using the GP0PUL register, it is possible to enable or disable pull-up resistors on the pins when they are configured as inputs. The pull-up resistors are automatically disabled when the pin is set as an output or when open circuit is enabled.

GPIO Data In

When configured as an input (by default), the GPIO input states are available in GP0IN.

Open Circuit Enable

If a pin is set as an output, setting the pin to open circuit disables the input paths. Open circuit is enabled by setting the corresponding bit on the GP0OCE register to Logic 1. External interrupts are not available when open circuit is enabled.

GPIO Data Out

When the GPIOs are configured as outputs, the values in GP0OUT are reflected on the GPIOs.

Bit Set

Bit set is used to set one or more GPIO data outputs without affecting others within a port. Only the GPIO corresponding with the write data bit equal to one is set. The remaining GPIOs are unaffected.

Bit Clear

Bit clear is used to clear one or more GPIO data outputs without affecting others within a port. Only the GPIO corresponding with the write data bit equal to one is cleared. The remaining GPIOs are unaffected.

Bit Toggle

Bit toggle is used to toggle one or more GPIO data outputs without affecting others within a port. Only the GPIO corresponding with the write data bit equal to one is toggled. The remaining GPIOs are unaffected.

GPIO Data Output Enable

The data output path is enabled. The values in GP0OUT are reflected on the GPIOs.

GPIO MEMORY MAPPED REGISTERS

Table 141. GPIO Port 0 Interface Memory Address (Base Address 0x40006000)

Offset	Name	Description	Access	Default
0x0000	GP0CON	GPIO configuration register	Read/write	0x0000
0x0004	GP0OEN	GPIO output enable register	Read/write	0x00
0x0008	GP0PUL	GPIO output pull-up enable register	Read/write	0x3F
0x000C	GP0OCE	GPIO open circuit enable register	Read/write	0x00
0x0014	GP0INR	GPIO input data register	Read	0xXX
0x0018	GP0OUT	GPIO data out register	Read/write	0x00
0x001C	GP0SET	GPIO data out set register	Write	0x00
0x0020	GP0CLR	GPIO data out clear register	Write	0x00
0x0024	GP0TGL	GPIO pin toggle	Write	0x00

GPIO Configuration Register

Address: 0x40006000, Reset: 0x0000, Name: GP0CON

For more information, see Table 140.

Table 142. GP0CON Register Bit Descriptions

Bits	Name	Description
[15:12]	Reserved	Reserved. These bits must be set to 0.
[11:10]	CON5	Configuration mode bits for P0.5.
[9:8]	CON4	Configuration mode bits for P0.4.
[7:6]	CON3	Configuration mode bits for P0.3.
[5:4]	CON2	Configuration mode bits for P0.2.
[3:2]	CON1	Configuration mode bits for P0.1.
[1:0]	CON0	Configuration mode bits for P0.0.

GPIO Output Enable Register

Address: 0x40006004, Reset: 0x00, Name: GP0OEN

Table 143. GP0OEN Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0OEN	Input or output enable. 0: enables the corresponding GPIO on Port 0 as an input. 1: enables the corresponding GPIO on Port 0 as an output.

GPIO Output Pull-Up Enable Register

0x40006008, Reset: 0x3F, Name: GP0PUL

Table 144. GP0PUL Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0PUL	Pull-up resistor enable. 0: disables the internal pull-up resistor on the corresponding GPIO on Port 0. 1: enables the internal pull-up resistor on the corresponding GPIO on Port 0.

GPIO Open Circuit Enable Register

Address: 0x4000600C, Reset: 0x00, Name: GP0OCE

Table 145. GP0OCE Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0OCE	Open circuit enable. Sets the GPIO pads on Port 0 to open-drain mode.

GPIO Input Data Register

Address: 0x40006014, Reset: 0xXX, Name: GP0INR

Contents of the GP0INR register depends on the digital level on the corresponding pins.

Table 146. GP0INR Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0IN	When read reflects the level (for example, 0 or 1) on the GPIO pins on Port 0 except when configured in open circuit

GPIO Data Out Register

Address: 0x40006018, Reset: 0x00, Name: GP0OUT

Table 147. GP0OUT Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0OUT	Data out bits. 0: cleared by user to drive the corresponding GPIO low. 1: set by user code to drive the corresponding GPIO high. Reads back the value on GPIO outputs. For example, writing GP0OUT = 0x12 drives 1 on P0.1 and P0.4, and the remaining GPIO are low, assuming these pins are configured as outputs.

GPIO Data Out Set Register

Address: 0x4000601C, Reset: 0x00, Name: GP0SET

Table 148. GP0SET Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0SET	Data out set bits. 0: no action. 1: set by user code to drive the corresponding GPIO high.

GPIO Data Out Clear Register

Address: 0x40006020, Reset: 0x00, Name: GP0CLR

Table 149. GP0CLR Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0CLR	Data out clear bits. 0: cleared by user code. Has no effect. 1: set by user code to drive the corresponding GPIO low.

GPIO Pin Toggle Register

Address: 0x40006024, Reset: 0x00, Name: GP0TGL

Table 150. GP0TGL Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
[5:0]	GP0TGL	Data out toggle bits. 0: cleared by user code. Clearing a bit has no effect on the corresponding state of the GPIO. 1: set by user code to invert the corresponding GPIO. Automatically cleared after input is toggled.

SERIAL PERIPHERAL INTERFACE

SPI FEATURES

The standard SPI include the following features:

- Serial clock phase mode (CPHA)
- Serial clock polarity mode (CPOL)
- LSB first transfer option
- Loopback mode
- Master or slave mode
- Transfer and interrupt mode
- Continuous transfer mode
- Transmit and receive first in, first out (FIFO)
- Interrupt mode, interrupt after one byte, two bytes, three bytes, or four bytes
- Receive overflow mode and transmit underrun mode
- Open circuit data output mode
- Full duplex communications supported

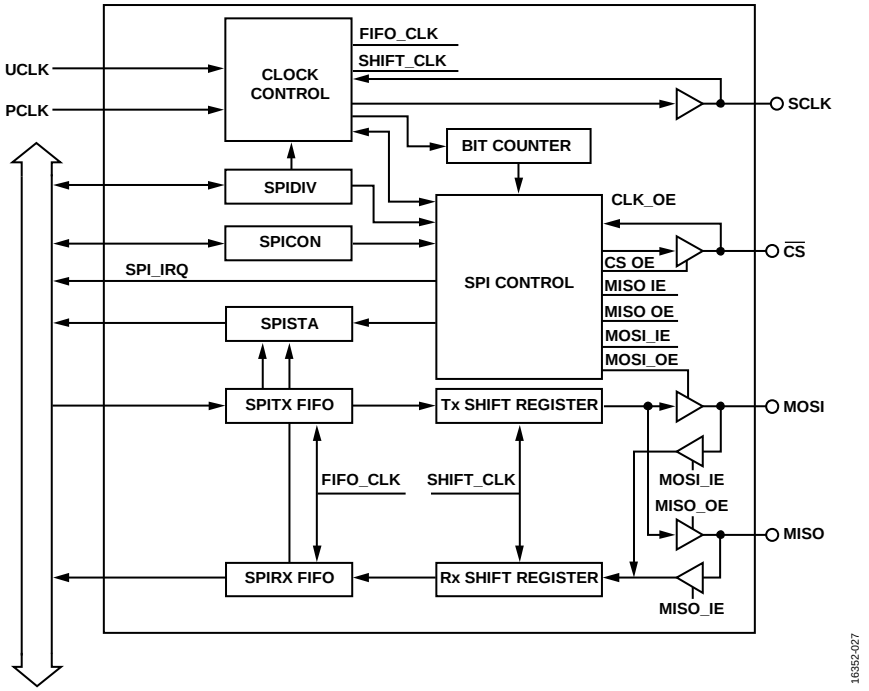

Figure 27. SPI Block Diagram

SPI OVERVIEW

The ADuCM300 integrates an SPI. The SPI is an industry-standard, synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, or full duplex. The SPI implemented on the ADuCM300 can operate at a maximum bit rate of 8 Mbps in master mode and slave mode.

SPI OPERATION

The SPI port can be configured for master or slave operation and consists of four pins: master in, slave out (MISO), master out, slave in (MOSI), serial clock input/output (SCLK), and chip select ($\overline{CS}$). In an SPI system, the master device generates a serial clock and asserts the chip select for the appropriate slave device. $\overline{CS}$ frames the transfer. Data is simultaneously transmitted from the master to the slave using the MOSI pin and transmitted from the slave to the master using the MISO pin. In normal master operation, the MOSI pin is an output and the MISO pin is an input. In normal slave operation, the MOSI pin is an input and the MISO pin is an output.

The GPIOs used for SPI communication must be configured in SPI mode before enabling the SPI peripheral, and the internal pull-up resistors on the SPI pins must be disabled via the GP0PUL register when using the SPI. To enable the SPI peripheral in slave mode, the $\overline{CS}$ pin must be monitored to ensure that communication is inactive first.

MISO 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.

MOSI 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.

SCLK Pin

The SCLK pin synchronizes the data being transmitted and received through the MOSI SCLK period. Therefore, a byte is transmitted or 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 SPICON register controls the polarity and phase of the clock, and the bit rate is defined in the SPIDIV register as follows:

$$f_{SERIALCLOCK} = f_{UCLK} \div (2 \times (1 + SPIDIV, Bits[5:0]))$$

The maximum data rate is 8 Mbps when using the hardware FIFO provided in the interface.

In slave mode, the SPICON 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 mode, data is transmitted on one edge of the SCLK signal and sampled on the other. Therefore, it is important that the polarity and phase be configured the same for the master and slave devices.

$\overline{CS}$ Pin

In SPI slave mode, a transfer is initiated by the assertion of $\overline{CS}$, 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 $\overline{CS}$. In slave mode, $\overline{CS}$is always an input.

In SPI master mode, $\overline{CS}$is an active low output signal. $\overline{CS}$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 (TIM) SPI bit (Register SPICON, Bit 6) determines the manner in which an SPI serial transfer is initiated. If the TIM SPI bit is set, a serial transfer is initiated after a write to the transmit FIFO occurs. If the transfer initiation mode bit is cleared, a serial transfer is initiated after a read of the receive FIFO. The read must be performed while the SPI interface is idle. A read performed during an active transfer cannot initiate another transfer.

For any setting of Register SPICON, Bit 1 and Register SPICON, Bit 6, the SPI simultaneously receives and transmits data. Therefore, during data transmission, the SPI is also receiving data and filling up the receive FIFO. If the data is not read from the receive FIFO, the overflow interrupt occurs after the FIFO starts to overflow. To prevent a read of the receive data or receive overflow interrupts, set Register SPICON, Bit 12 and the receive data is not saved to the receive FIFO.

Similarly, to receive only data and not write data to the transmit FIFO, set Register SPICON, Bit 13 to avoid receiving underrun interrupts from the transmit FIFO.

Transmit Initiated Transfer

For transfers initiated by a write to the transmit FIFO (SPITX), the SPI begins transmitting as soon as the first byte is written to the FIFO, irrespective of the configuration in Register SPICON, Bits[15:14]. The first byte is immediately read from the FIFO and written to the transmit shift register, and the transfer commences.

If the continuous transfer enable bit (Register SPICON, Bit 11) is set, the transfer continues until no valid data is available in the transmit FIFO. There is no stall period between transfers where $\overline{CS}$ is deasserted. $\overline{CS}$ is asserted and remains asserted for the duration of the transfer until the transmit FIFO is empty. The time the transfer ends does not depend on Register SPICON, Bits[15:14]. 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 the continuous transfer enable bit is cleared, each transfer consists of a single, 8-bit serial transfer. If valid data exists in the transmit FIFO, a new transfer is initiated after a stall period with $\overline{CS}$ being deasserted.

Receive Initiated Transfer

Transfers initiated by a read of the receive FIFO (SPIRX) depend on the number of bytes to be received in the FIFO. If Register SPICON, Bits[15:14] are set to 3, and a read to the receive FIFO occurs, the SPI initiates a 4-byte transfer. If continuous mode is set, the four bytes occur continuously with no deassertion of $\overline{CS}$ between bytes. If continuous mode is not set, the four bytes occur with stall periods between transfers with the $\overline{CS}$ pin being deasserted.

If Register SPICON, Bits[15:14] are set to 0x2 and a read to the receive FIFO occurs, the SPI initiates a 3-byte transfer. If Register SPICON, Bits[15:14] are set to 0x1 and a read to the receive FIFO occurs, the SPI initiates a 2-byte transfer. If Register SPICON, Bits[15:14] are set to 0x0 and a read to the receive FIFO occurs, the SPI initiates a 1-byte transfer.

A read of the receive 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 $\overline{CS}$. The device as a slave transmits and receives 8-bit data until the transfer is concluded by the deassertion of $\overline{CS}$.

The SPI transfer protocol diagrams (see Figure 28 and Figure 29) illustrate the data transfer protocol for the SPI and the effects of the CPHA (Register SPICON, Bit 2) bit and CPOL (Register SPICON, Bit 3) bit in the control register on that protocol.

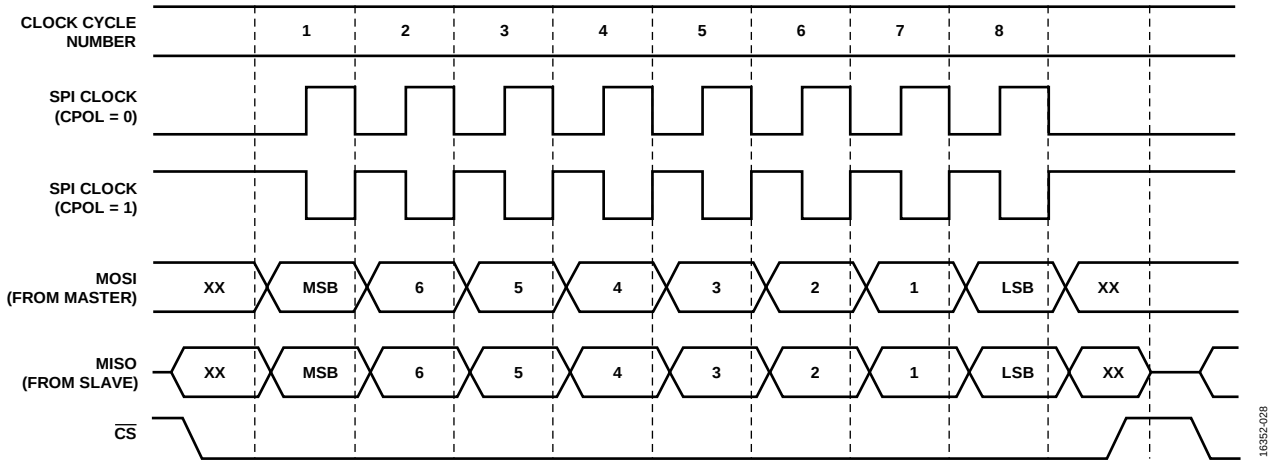

Figure 28. SPI Transfer Protocol, CPHA = 0

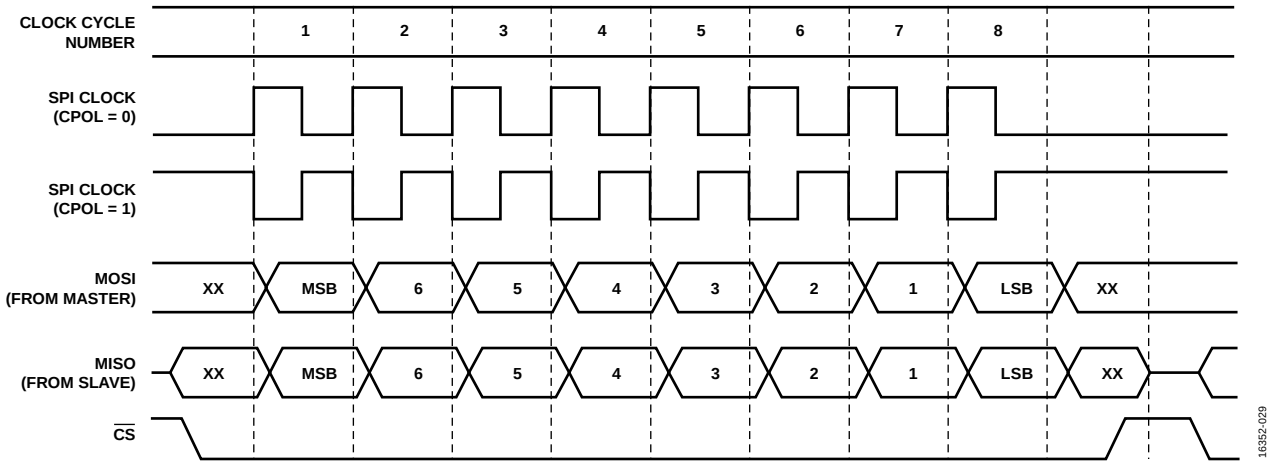

Figure 29. SPI Transfer Protocol, CPHA = 1

SPI Data Underrun and Overflow

If the transmit underrun mode bit, zero transmit enable (ZEN), Register SPICON, Bit 7, is cleared, the last byte from the previous transmission 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 receive overflow overwrite enable bit, RXOF (Register SPICON, Bit 8), is set, the valid data in the receive 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 receive FIFO, the oldest byte is overwritten first, followed by the next oldest byte, and so on.

SPI INTERRUPTS

There are five sources of interrupts. Register SPISTA, Bit 0 reflects the state of the interrupt line. Register SPISTA, Bits[7:4] and Register SPISTA, Bit 12 reflect the state of the five sources. The SPI generates either a transmit IRQ or a receive IRQ. Both interrupts cannot be enabled at the same time. The appropriate interrupt is enabled using the TIM bit in the SPICON register. If TIM = 1, a transmit IRQ is enabled. If TIM = 0, a receive IRQ is enabled. In addition, the CSERR bit (Register SPISTA, Bit 12) is always generating an interrupt.

In master mode, interrupts are generated after the last SCLK edge for each byte transfer. In slave mode, interrupts are generated either by the first SCLK edge of the next byte transferred or by the deassertation of $\overline{CS}$.

Transmit Interrupt

If the TIM bit (Register SPICON, Bit 6) is set, the transmit FIFO status causes the interrupt. Register SPICON, Bits[15:14] control when the interrupt occurs, as shown in Table 151.

The interrupts are generated depending on the number of bytes transmitted and not on the number of bytes in the FIFO, unlike the receive interrupt, which depends on the number of bytes in the receive FIFO and not on the number of bytes received.

The transmit interrupt is cleared by a read to the status register. The status of this interrupt can be cleared by reading Register SPISTA, Bit 5. The interrupt is disabled if Register SPICON, Bit 13 is left high.

A write to the control register, SPICON, resets the transmitted byte counter back to zero. For example, when Register SPICON, Bits[15:14] are set to 0x3 and SPICON is written to after three bytes are transmitted, the transmit interrupt does not occur until another four bytes are transmitted.

Table 151. Register SPICON, Bits[15:14] IRQ Mode Bits

Register SPICON, Bits[15:14]	Interrupt Condition
0x0	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.
0x1	An interrupt is generated after every two bytes that are transmitted.
0x2	An interrupt occurs after every third byte that is transmitted.
0x3	An interrupt occurs after every fourth byte that is transmitted.

Receive Interrupt

If the TIM bit is cleared, the receive FIFO status causes the interrupt. Register SPICON, Bits[15:14] control when the interrupt occurs. The interrupt is cleared by a read of SPISTA. The status of this interrupt can be cleared by reading Register SPISTA, Bit 6.

Interrupts are generated only when data is written to the FIFO. For example, if Register SPICON, Bits[15:14] are 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 cannot be regenerated. Another interrupt cannot be generated until another byte is received into the FIFO.

The interrupt depends on the number of valid bytes in the FIFO and not on the number of bytes received. For example, when Register SPICON, Bits[15:14] are 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 Register SPICON, Bit 12 is left high.

Underrun and Overflow Interrupts

Register SPISTA, Bit 7 and Register SPISTA, Bit 4 also generate SPI interrupts. When a transfer starts with no data in the transmit FIFO, Register SPISTA, Bit 4 is set to indicate an underrun condition. This condition causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of Register SPICON, Bits[15:14]. This interrupt is disabled if Register SPICON, Bit 13 is set. At the last edge of the SPICLK of the actual byte transferred, the next byte must be available in the transmit FIFO to be transferred into the transmit shift register.

When data is received and the receive FIFO is already full, Bit 8 of the status register goes high to indicate 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 Register SPICON, Bits[15:14]. This interrupt is disabled if Register SPICON, Bit 12 is set.

All interrupts are cleared by a read of the status register or if Register SPICON, Bit 0 is deasserted. The receive interrupt and transmit interrupt are also cleared if the relevant flush bits are asserted. Otherwise, the interrupts stay active even if the SPI is reconfigured.

WIRE-OR'ED MODE (WOM)

To prevent contention when the SPI is used in a multimaster or multislave system, the MOSI and MISO data output pins 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 configuration register controls the pin enable outputs for the data lines. With WOM enabled in master mode, the output drive is enabled when a 0 is being transmitted on the MOSI pin, and the output driver is disabled when a 1 is being transmitted on the MOSI pin. The line is pulled up by the pull-up resistor.

Similarly, in slave mode and with WOM enabled, the output drive is enabled when a 0 is being transmitted on the MISO pin, and the output driver is disabled when a 1 is being transmitted on the MISO pin.

CSERR CONDITION

The SPI bit counter (internal) (see Figure 27) is reset after the completion of eight clocks of SCLK. The CSERR bit (Register SPISTA, Bit 12) indicates if an erroneous deassertion of the $\overline{CS}$ signal has been detected before the completion of all the eight SCLK cycles. This bit generates an interrupt. The bit counter stays at the value where it stopped and then continues from there when $\overline{CS}$ is asserted afterward. This continuation of the bit counter may cause inconsistent data transfers. To avoid inconsistent data transfers, the CSERR detection circuit is available. The CSERR detection circuit checks for $\overline{CS}$ deassertion when bit counter is not equal to 7 (reset value). If the condition is met, the circuit asserts a CSERR signal. If an interrupt occurs, generated by the CSERR bit (Register SPISTA, Bit 12), the SPI enable bit (Register SPICON, Bit 0) must be disabled and restarted to enable a clean recovery. This process ensures that subsequent transfers are error free. The BCRST bit (Register SPIDIV, Bit 7) must be set at all times in both slave mode and master mode, except when a midbyte stall in SPI communication is required. In this case, the CSERR flag is set, but can be ignored.

The SPI must only be reenabled when the signal is high.

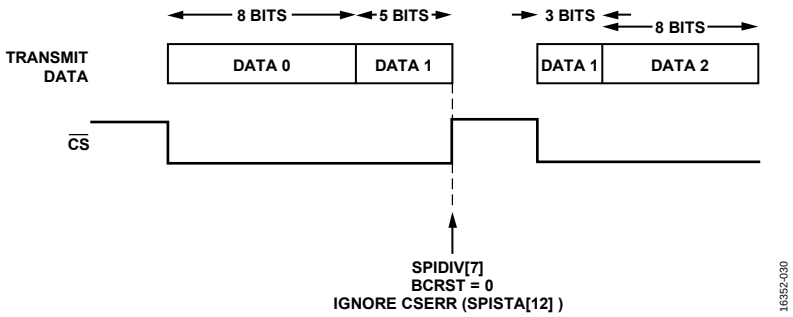

Figure 30. High Voltage Interface, Top Level Block Diagram

SPI AND POWER-DOWN MODES

In master mode, before entering hibernate mode or system halt mode, it is recommended to disable the SPI block in Register SPICON, Bit 0. In slave mode, 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. In slave mode, the SPI must be reenabled only if the $\overline{CS}$ line is high.

SPI MEMORY MAPPED REGISTERS

Table 152. SPI Peripheral Memory Address (Base Address 0x40004000)

Offset	Name	Description	Access	Default
0x0000	SPISTA	SPI status register	Read	0x0000
0x0004	SPIRX	SPI 8-bit receive register	Read	0x0000
0x0008	SPITX	SPI 8-bit transmit register	Write	0x0000
0x000C	SPIDIV	SPI 8-bit baud rate selection register	Read/write	0x0000
0x0010	SPICON	SPI 16-bit configuration register	Read/write	0x0000

SPI Status Register

Address: 0x40004000, Reset: 0x0000, Name: SPISTA

Table 153. SPISTA Register Bit Descriptions

Bits	Name	Description
[15:13]	Reserved	Reserved.
12	CSERR	Detected an abrupt $\overline{CS}$ deassertion. 0: cleared when the SPISTA register is read. 1: set when the line $\overline{CS}$ is deasserted abruptly, even before the full byte of data is transmitted completely. This bit causes an interrupt. If the CSERR bit is set, it is recommended to clear the enable bit in the SPICON register to ensure a clean recovery.
11	RXS	SPI receive FIFO excess bytes present. Indicates when there are more bytes in the receive FIFO than the receive interrupt indicated. This bit depends on Register SPICON, Bit[15:14]. 00: RXS is set if there are two or more bytes in the receive FIFO. 01: RXS is set if there are three or more bytes in the receive FIFO. 10: RXS is set if there are four or more bytes in the receive FIFO. 11: RXS is not set. Cleared to 0 when the number of bytes in the FIFO is equal or less than the number of bytes in Register SPICON, Bits[15:14]. This bit is not cleared when the SPISTA register is read. This bit is not dependent on Register SPICON, Bit 6 and does not cause an interrupt.
[10:8]	RXFSTA	SPI receive FIFO status bits. 000: FIFO is empty. 001: one valid byte in the FIFO. 010: two valid bytes in the FIFO. 011: three valid bytes in the FIFO. 100: four valid bytes in the FIFO.
7	RXOF	SPI receive FIFO overflow status bit (interrupt). 0: cleared to 0 when the SPISTA register is read. 1: set to 1 when the receive FIFO is already full when new data is loaded to the FIFO. This bit generates an interrupt, except when the RFLUSH bit is set in the SPICON register.
6	RX	SPI receive IRQ status bit. 0: cleared to 0 when the SPISTA register is read. 1: set to 1 when a receive interrupt occurs. This bit is set when the TIM bit in the SPICON register is cleared, and the required number of bytes are received.
5	TX	SPI transmit IRQ status bit. 0: cleared to 0 when the SPISTA register is read.
		1: set to 1 when a transmit interrupt occurs. This bit is set when the TIM bit in the SPICON register is set, and the required number of bytes are transmitted.
4	TXUR	SPI transmit FIFO underrun (interrupt). 0: cleared to 0 when the SPISTA register is read. 1: set to 1 when a transmit is initiated without any valid data in the transmit FIFO. This bit generates an interrupt, except when the TFLUSH bit is set in the SPICON register.

Bits	Name	Description
[3:1]	TXFSTA	Indicates how many valid bytes are in the SPI transmit FIFO. 000: FIFO is empty. 001: one valid byte in the FIFO. 010: two valid bytes in the FIFO. 011: three valid bytes in the FIFO. 100: four valid bytes in the FIFO. In master mode, when a byte is copied from the FIFO to the shift register, the status bits are decremented. In slave mode, the status bits are not decremented until the byte is transmitted out of the shift register. Therefore, in master mode, the status bits reflect the number of valid bytes in the FIFO. However, in slave mode, the status bits reflect the number of valid bytes in the FIFO and in the shift register. This behavior is due to a byte being immediately transmitted in master mode, whereas in slave mode, a valid byte can wait in the shift register for the master SPI to initiate a transfer.
0	IRQ	SPI interrupt status bit. 0: cleared to 0 after reading the SPISTA register. 1: set to 1 when an SPI-based interrupt occurs.

SPI Receive Register

Address: 0x40004004, Reset: 0x0000, Name: SPIRX

Table 154. SPIRX Register Bit Descriptions

Bits	Name	Description
[15:8]	Reserved	Reserved. These bits must be set to 0.
[7:0]	Value	8-bit receive register. A read of the receive FIFO returns the next byte to be read from the FIFO. A read of the FIFO when it is empty returns 0s.

SPI Transmit Register

0x40004008, Reset: 0x0000, Name: SPITX

Table 155. SPITX Register Bit Descriptions

Bits	Name	Description
[15:8]	Reserved	Reserved. These bits must be set to 0.
[7:0]	Value	8-bit transmit register. A write to the transmit FIFO address space writes data to the next available location in the transmit FIFO. If the FIFO is full, the oldest byte of data in the FIFO is overwritten. A read from this address location returns 0s.

SPI Baud Rate Selection Register

Address: 0x4000400C, Reset: 0x0000, Name: SPIDIV

When setting the SPI serial clock, the PCLK frequency must be taken into account. The PCLK frequency can be no less than half the SPI serial clock frequency. For example, if the SPI clock divide register is set to 0x0000 (SCLK frequency = ½ UCLK frequency), the maximum that the CD bits can be set to is 2 (PCLK frequency = ¼ UCLK frequency).

Table 156. SPIDIV Register Bit Descriptions

Bits	Name	Description
[15:8]	Reserved	Reserved. These bits must be set to 0.
7	BCRST	Reset mode for CSERR. This bit is used to configure the expected behavior of the SPI interface logic after an abrupt deassertion of $\overline{CS}$. 0: SPI interface logic continues from where it stopped. The SPI can receive the remaining bits when $\overline{CS}$ is asserted and user code must ignore the CSERR interrupt. 1: SPI interface logic is reset after a CSERR condition and user code must clear the SPI enable bit in the SPICON register.
6	Reserved	Reserved. This bit must be set to 0.
[5:0]	DIV	Factor used to divide UCLK to generate the serial clock. $f_{SERIAL\ CLOCK} = f_{UCLK} \div (2 \times (1 + SPIDIV, Bits[5:0]))$ Maximum frequency for the serial clock is ½ the UCLK frequency. These bits are only used for master mode. In slave mode, there is no need to set the serial clock frequency. The slave receives the clock from the master.

SPI Configuration Register

Address: 0x40004010, Reset: 0x0000, Name: SPICON

When changing the configuration, take care not to change it during a data transfer to avoid corrupting the data. It is recommended to change the configuration when the module is disabled (disable the SPI, enable = 0), then reconfigure and reenable the SPI (enable = 1). When reconfiguring from slave mode to master mode, or vice versa, both FIFOs must be empty. It is recommended in slave mode to reenable the SPI only if the $\overline{CS}$ line is high.

Table 157. SPICON Register Bit Descriptions

Bits	Name	Description
[15:14]	MOD	SPI IRQ mode bits. When the TIM bit is set, these bits configure when the transmit interrupt or receive interrupt occurs in a transfer. 00: transmit interrupt occurs when one byte is transmitted. Receive interrupt occurs when one or more bytes have been received into the FIFO. 01: transmit interrupt occurs when two bytes are transmitted. Receive interrupt occurs when two or more bytes have been received into the FIFO. 10: transmit interrupt occurs when three bytes are transmitted. Receive interrupt occurs when three or more bytes have been received into the FIFO. 11: transmit interrupt occurs when four bytes are transmitted. Receive interrupt occurs when the receive FIFO is full, or four bytes are present.
13	TFLUSH	SPI transmit FIFO flush enable bit. 0: disable transmit FIFO flushing. 1: flush the transmit FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is left high, either the last transmitted value or 0x00 is transmitted, depending on the ZEN bit. Any writes to the transmit FIFO are ignored while this bit set.
12	RFLUSH	SPI receive FIFO flush enable bit. 0: disable receive FIFO flushing. 1: flush the receive 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 receive FIFO initiates a transfer.
11	CON	Continuous transfer enable bit. 0: 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. The $\overline{CS}$ line is deactivated for this one serial clock cycle. 1: enable continuous transfer. In master mode, the transfer continues until no valid data is available in the SPITX register. $\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until SPITX is empty.
10	LOOPBACK	Loopback enable bit. 0: normal mode. 1: connect MISO to MOSI. Data transmitted from the SPITX register is looped back to the SPIRX register. The MASEN bit must be set for loopback mode to work.
9	OEN	Slave MISO output enable bit. 0: disable the output driver on the MISO pin. The MISO pin is open circuit when this bit is clear. 1: MISO operates as normal.
8	RXOF	SPIRX overflow overwrite enable bit. 0: the new serial byte received is discarded. 1: the valid data in the SPIRX register is overwritten by the new serial byte received.
7	ZEN	SPI transmit 0s when the transmit FIFO is empty. 0: transmit the last transmitted value when there is no valid data in the transmit FIFO. 1: transmit 0x00 when there is no valid data in the transmit FIFO.
6	TIM	SPI transfer and interrupt mode. 0: initiate transfer with a read of the SPIRX register. The read must be done while the SPI interface is idle. Interrupt occurs only when the receiver is full. 1: initiate transfer with a write to the SPITX register. Interrupt occurs only when the transmitter is empty.
5	LSB	LSB first transfer enable bit. 0: MSB is transmitted first. 1: LSB is transmitted first.

Bits	Name	Description
4	WOM	SPI wire-OR'ed mode enable bit. 0: normal output levels. 1: enable open circuit data output. Master mode: when a 0 is being transmitted on the MOSI pin, the output driver is enabled. When a 1 is being transmitted on the MOSI pin, the output driver is disabled, and an external pull-up resister is required to pull the pin high. Typical resistor value is 1 kΩ. Slave mode: when a 0 is being transmitted on the MISO pin, the output driver is enabled. When a 1 is being transmitted on the MISO pin, the output driver is disabled, and an external pull-up resister is required to pull the pin high. Typical resistor value is 1 kΩ.
3	CPOL	Serial clock polarity mode bit. 0: serial clock idles low. 1: serial clock idles high.
2	CPHA	Serial clock phase mode bit. 0: serial clock pulses at the middle of the first data bit transfer. 1: serial clock pulses at the start of the first data bit.
1	MASEN	Master mode enable bit. 0: enable slave mode. 1: enable master mode.
0	Enable	SPI enable bit. 0: disable the SPI. Clearing this bit also resets all the FIFO related logic and bit counter to enable a clean start. 1: enable the SPI.

HIGH VOLTAGE PERIPHERAL CONTROL INTERFACE

HIGH VOLTAGE PERIPHERAL CONTROL INTERFACE OVERVIEW

A high voltage peripheral is connected to the low voltage die via a serial interface. The low voltage die is the serial interface master. The serial interface is clocked with an interdie communication clock (DCLK, UCLK/8).

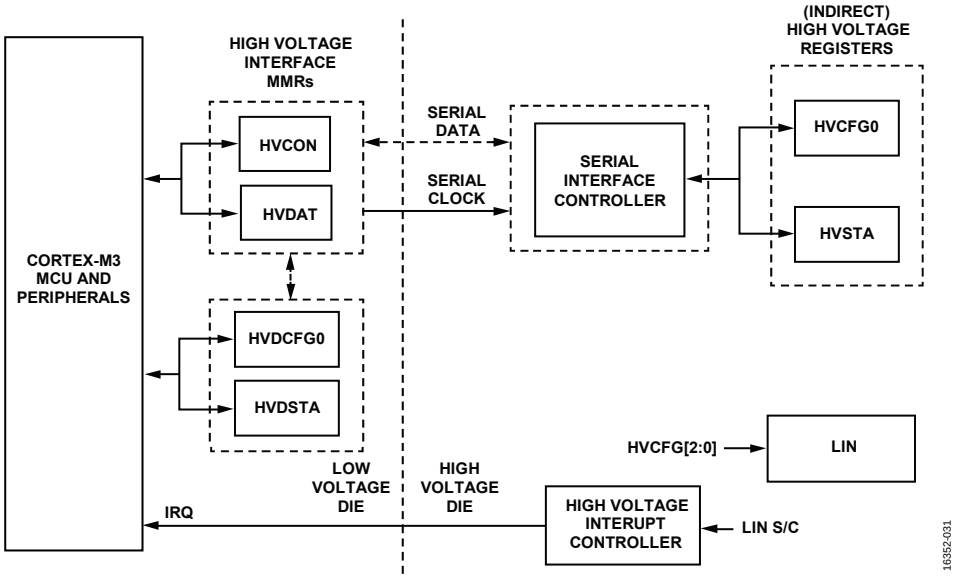

Figure 31. High Voltage Interface, Top Level Block Diagram

HIGH VOLTAGE PERIPHERAL CONTROL INTERFACE OPERATION

The two following methods are implemented for communicating data to and from the top die of the ADuCM300:

- An indirect high voltage register access method also known as the keyhole approach.
- A direct MMR programming method.

Indirect Approach

To read data or write data to the high voltage top die, an indirect high voltage register access method is implemented in the master side of the interface. This means that the user must use the HVDAT register and the HVCON register on the bottom die for accessing the registers on the top die.

A high voltage register access is initiated by writing the appropriate high voltage command into the HVCON register. After writing the appropriate high voltage command into the HVCON register, the HVCON register contains information on the progress and validity of the high voltage register access. After a high voltage register is accessed, the HVDAT register contains the last executed command in Register HVDAT, Bits[11:8].

The content of the HVDAT register after an executed high voltage command is as follows:

- High voltage read: after the high voltage register read is completed, the user can read the high voltage register data from the low byte of the HVDAT register.
- High voltage write: the data to be written to the high voltage register has to be programmed to the HVDAT register prior to initiating a high voltage register write.

On a high voltage interrupt event, a high voltage read of the HVSTA register is automatically initiated. If the high voltage interface is busy, the automatic HVSTA read commences when the current command completes. After the HVSTA read completes, an interrupt is issued to the core. Read the HVCON register to check the progress of the high voltage register access and check Register HVDAT, Bits[11:8] to confirm that the HVSTA read has been performed.

Direct MMR Programming

This method uses the HVDCFG0 register to write data to the HVCFG0 register on the high voltage top die. Writing a value to the HVDCFG0 register automatically initiates a write to the HVCFG0 register on the high voltage die. The HVDCFG0 register on the low voltage MMR is then automatically updated to the readback value from the top die.

The HVDSTA register on the low voltage die is a shadowed register of the HVSTA register on the high voltage die. On a high voltage interrupt, an HVSTA register read is automatically initiated and the shadowed HVDSTA is updated accordingly. An HVDSTA register update only occurs with a high voltage interrupt or when an HVSTA register read via the indirect high voltage register access method is performed.

When a high voltage interrupt occurs and the automatically initiated HVSTA read is not completed, the internal hardware associated with the shadowed MMRs reiterates an HVSTA read command until an HVSTA read has been performed. In case of a fault within a high voltage peripheral control interface, this feature prevents any further high voltage communication.

Therefore, it is recommended to implement a timeout counter functionality within the high voltage communication user code, which indicates a high voltage communication fault.

A delay of 20 µs is required when using the direct high voltage register access method to ensure that data has been transferred correctly. To avoid this delay, the user can poll the HVCON register to check if the information has been written correctly.

HIGH VOLTAGE DIE REGISTERS

The registers residing on the high voltage die can only be accessed using the high voltage peripheral control interface.

Table 158. High Voltage Die Registers

Name	Description	Access	Default
HVCFG0	High voltage configuration register (accessible with indirect high voltage register access method)	Read/write	0x00
HVSTA	High voltage status register (accessible with indirect high voltage register access method)	Read	0x00

High Voltage Configuration Register

Address: not applicable, indirectly addressed register, Reset: 0x00, Name: HVCFG0

The HVCFG0 register is located on the high voltage die and can be accessed via the indirect high voltage register access method and direct MMR programming method.

The HVCFG0 register is used to enable the LIN hardware, attenuator diagnostic features.

Table 159. HVCFG0 Register Bit Descriptions

Bits	Name	Description
[7:6]	Reserved	Reserved. These bits must be set to 0.
5	ASEL	Voltage attenuator selection. 0: 1:24 attenuator connected to ADC1 input (default). 1: 1:48 attenuator connected to ADC1 input.
4	Reserved	Reserved. These bits must be set to 0.
3	VE	AIN4 attenuator diagnostic current source enable bit. 0: disable the AIN4 attenuator diagnostic function. 1: enable the AIN4 attenuator diagnostic function.
2	IOE	LIN driver reenable. If the current into or through the LIN pin exceeds $I_{LIN\_DOM\_MAX}$, while in the dominant state, the LIN bus driver is disabled. The user writes to this bit to reenable the LIN driver after an event such as this. 0: automatically cleared by hardware after being set. 1: reenable the LIN bus driver.
1	DLSCP	Disable LIN short-circuit protection circuitry. 0: allow short-circuit detection, and the LIN driver automatically disables. 1: disable the LIN driver autodisabling functions. High voltage interrupt still issued in short-circuit detection.
0	LIN_EN	LIN transmit enable. 0: clear to 0 to disable the top die LIN transmitter. 1: set to 1 to enable the top die LIN transmitter.

High Voltage Status Register

Address: not applicable, indirectly addressed register, Reset: 0x00, Name: HVSTA

The HVSTA register is used to identify the high voltage hardware interrupt conditions.

Table 160. HVSTA Register Bit Descriptions

Bits	Name	Description
[7:4]	Revision ID	Revision ID bits for the high voltage die.
[3:2]	Reserved	Reserved.
1	LSCS	LIN short-circuit status. 0: cleared to 0 if short-circuit condition is removed. 1: set to 1 if the current into or through the LIN pin exceeds $I_{LIN\_DOM\_MAX}$, while in dominant state, indicative of a short-circuit to VDD.
0	LSCI	LIN short-circuit interrupt. 0: cleared to 0 if the HVSTA register is read. 1: set to 1 if the current through the LIN pin exceeds $I_{LIN\_DOM\_MAX}$, while in dominant state, indicative of a short-circuit to VDD.

HIGH VOLTAGE PERIPHERAL INTERFACE MEMORY MAPPED REGISTERS

Table 161. High Voltage Peripheral Interface Memory Mapped Registers (Base Address 0x40003000)

Offset	Name	Description	Access	Default
0x0000	Reserved	User must not write to this location	Read	0x00
0x0004	HVCON	High voltage peripheral control interface command register (used with indirect high voltage register access method)	Read/write	0x0000
0x0008	Reserved	User must not write to this location	Read	0x00
0x000C	HVDAT	High voltage data register (used with indirect high voltage register access method)	Read/write	0x0800
0x0010	HVDCFG0	Shadowed high voltage configuration register (used with direct high voltage register access method)	Read/write	0x00
0x0018	HVDSTA	Shadowed high voltage status register (used with direct high voltage register access method)	Read	0x00
0x001C	Reserved	User must not write to this location	Read	0x00

High Voltage Peripheral Control Interface Command Register

Address: 0x40003004, Reset: 0x0000, Name: HVCON

If the user is using the keyhole approach, the HVCON register and HVDAT registers are used together.

To write data to the HVCFG0 register on the top die, first data is placed into the HVDAT register, and then 0x08 is written into the HVCON register.

To read data from the top die registers (HVCFG0 or HVSTA), 0x00 or 0x02 is placed in the HVCON register. A data read is then triggered from the top die, and the selected register contents are copied down into the HVDAT register. Care must be taken to observe the relevant busy, P_BIT, and ACK_BIT bits of the HVCON register.

The HVCON register accepts the following commands, as shown in Table 162, when written to.

When read, the HVCON register returns a 4-bit value.

Table 162. HVCON Commands

Group	Command	Description
Read Commands	0x00	Read back high voltage register HVCFG0 into HVDAT
	0x02	Read back high voltage register HVSTA into HVDAT
Write Commands	0x08	Write to high voltage register HVCFG0 from HVDAT
	Other	Reserved

Table 163. HVCON Register Bit Descriptions

Bits	Name	Description
[31:4]	Reserved	Reserved. These bits must be set to 0.
3	MMRMODE_BUSY	0: no mirrored MMR write in progress. Previous mirrored MMR writes are completed. 1: mirrored MMR transaction in progress. Do not write to high voltage die using keyhole method when this bit is set to 1.
2	ACK_BIT	0: high voltage slave transmit not acknowledge (NACK). 1: high voltage slave transmit acknowledge (ACK).
1	P_BIT	0: receive parity failed. 1: receive parity achieved.
0	Busy	1: indicates communication in progress. Further commands must not be issued. 0: indicates communication not in progress.

High Voltage Data Register

Address: 0x4000300C, Reset: 0x0800, Name: HVDAT

Register HVDAT, Bits[7:0] store the data read from or to be written to the top die.

Register HVDAT, Bits[11:8] store the last performed high voltage access command.

Table 164. HVDAT Register Bit Descriptions

Bits	Name	Description
[31:12]	Reserved	Reserved. These bits must be set to 0.
[11:8]	COM	Current command. Contains the last write to HVCON.
[7:0]	Data	Value of shift register in the communications. When transmissions occur, data is stored here.

Shadowed High Voltage Configuration Register

Address: 0x40003010, Reset: 0x00, Name: HVDCFG0

HVDCFG0 is a shadowed MMR of the HVCFG0 register on the top die. Bit locations are identical.

Shadowed High Voltage Status Register

Address: 0x40003018, Reset: 0x00, Name: HVDSTA

HVDSTA is a shadowed MMR of the HVSTA register on the top die. Bit locations are identical.

LIN INTERFACE

LIN OVERVIEW

The ADuCM300 implements a low overhead LIN interface that is compliant with LIN 1.3 and LIN 2.2/SAE J2602-2 specifications. The device operates as a slave only interface, operating from 1 kbps to 20 kbps. All lower rates are interpreted as 1 kbps.

The interface consists of a two-die solution. As shown in Figure 32, an integrated LIN transceiver is present on the top die that communicates directly to the LIN logic on the bottom die. The logic uses three internal Cortex-M3 interrupts, LININT0, LININT1, and LININT2, to signify to the core that a LIN event has occurred.

LIN FEATURES

The following features are incorporated in the LIN interface:

- Efficient LIN data handling with storage for up to eight data bytes
- Built in parity check for protected identifier (PID)
- Framing error checking
- Running checksum (classic or enhanced) on received bytes
- Partial PID decoding to determine diagnostic master request frame
- Break symbol detection at any time
- Collision detection
- UART mode
- Automatic transceiver delay time compensation

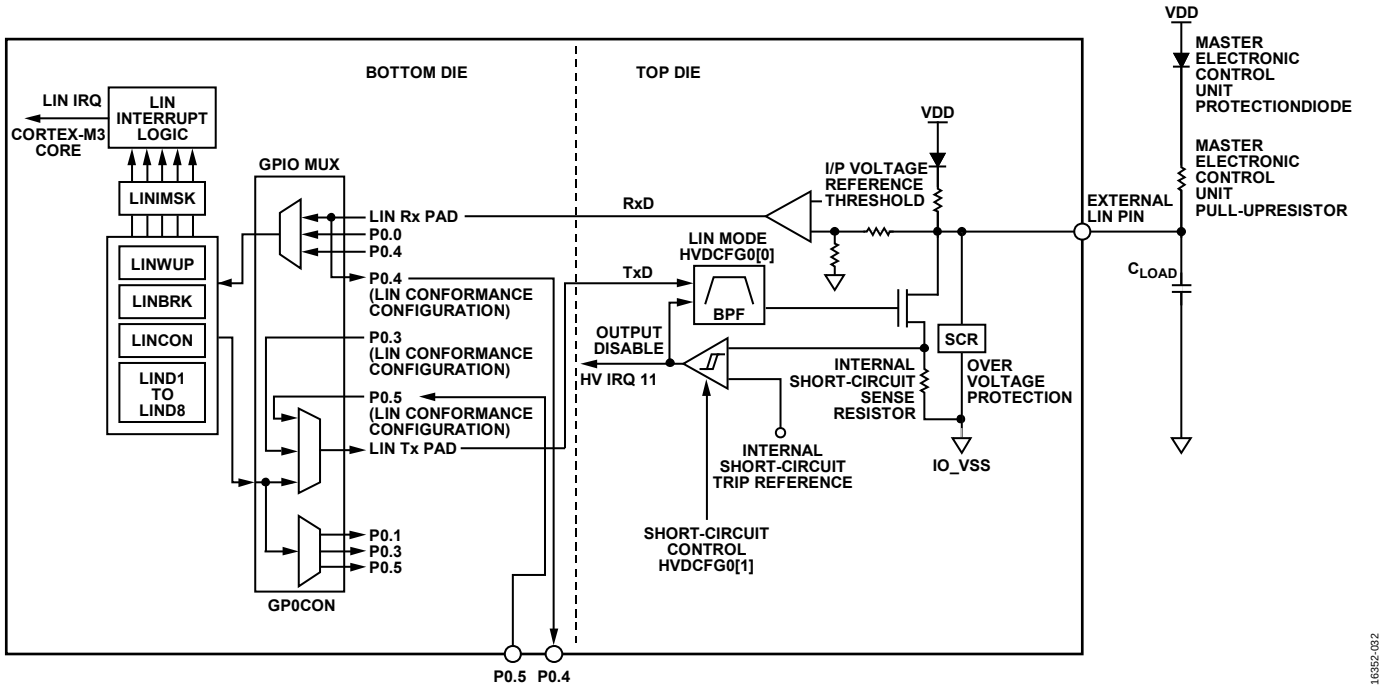

Figure 32. LIN Block Diagram

LIN USER OPERATION

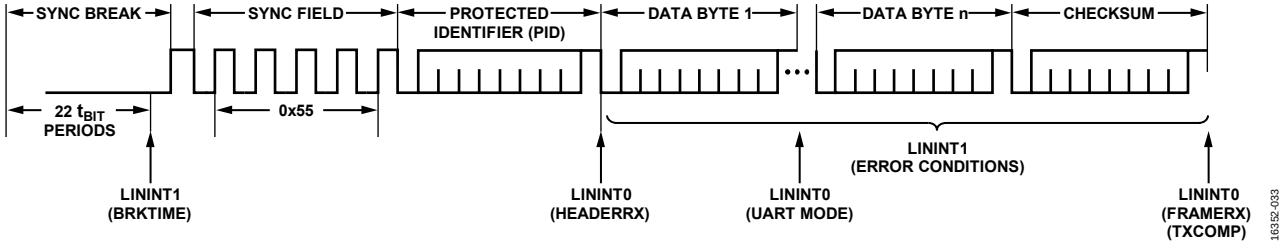

Figure 33. LIN Interrupts

LIN Transmission

A maskable software interrupt (LININT0) occurs on the reception of a PID. After identifying the PID, the user code determines if the device must respond, receive, or ignore the PID. If a transmission is required, the user code must perform the following:

- Set the number of bytes to transmit in the LINCNT register.
- Place all of the data bytes to transmit into their individual LINDx registers.
- Set Register LINCON, Bit 1 to 1 to signify begin transmission.

When the number of data bytes transmitted is equal to the value held in LINCNT, the checksum (which is automatically calculated) is then transmitted. After the checksum is transmitted, the frame is considered complete, and the LININT0 interrupt is set again.

LIN Reception

The LIN interface automatically changes to receive mode on reception of a break symbol.

The LININT0 interrupt indicates that a PID has been received. If the software decides that full message reception is required, the expected number of bytes is placed in the LINCNT register, and the interface begins to receive all the data bytes from the master. When all expected bytes are received, the checksum calculation is started, and if the calculation is correct, LININT0 is triggered again.

If the software determines that the PID does not require a response, the LINCNT register must be set to 0 indicating to the hardware to ignore the rest of the frame.

With back to back LIN transmission, a minimum delay of one t_{BIT} period between frames is required to ensure no frames are missed. If a break arrives in the middle of a frame, the break is detected. However, if a frame completes, there is a period of time, less than 1 t_{BIT} length, after the frame when a break is not detected and a frame may be missed.

Sleep and Wake Functionality

The ADuCM300 LIN interface has integrated LIN sleep and wake functionality. The LININT2 interrupt is used to notify the user that a wake or sleep event has occurred. When the LININT2 interrupt is triggered, the user must interrogate the LINSTA register to determine the source of the interrupt.

The two LININT2 interrupt sources are as follows:

- The LIN interface has measured a period of inactivity on the LIN bus corresponding to the LINSLP register setting. The LINSLP register allows flexible configuring of the bus idle timeout time from 0.5 sec to 16 sec.

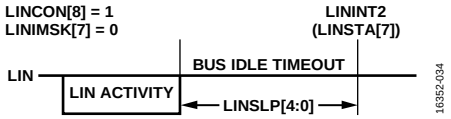

Figure 34. Sleep Interrupt

- The LIN interface has received a LIN wake-up frame during power-down mode. A valid wakeup is a dominant state on the LIN bus for a time set by the LINWUP register, followed by a rising edge. The LIN bus must remain high for at least two 32.768 kHz clock periods for the LININT2 interrupt to be generated.

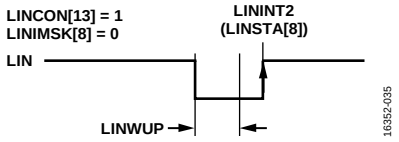

Figure 35. Wake-Up Interrupt

LIN Diagnostic Frame Support

The ADuCM300 LIN interface has intelligent support for diagnostic command frames (PID 0x3C) from a LIN master. In the event a 0x3C PID is sent from the LIN master and Register LINCON, Bit 14 is cleared to 0, the ADuCM300 cannot generate a LININT0 PID interrupt. The ADuCM300 then generates a LININT0 frame received interrupt only on the following conditions:

- Byte 0 contains the node address for diagnostic (NAD).
- Byte 0 contains 0x00 (sleep frame).
- Byte 0 contains 0x7F (broadcast frame).

All other conditions cause the entire LIN diagnostic frame to be ignored, and no interrupt is generated on the LIN frame. This feature means there is minimum software intervention in servicing LIN diagnostic frames.

This feature can be disabled by setting Register LINCON, Bit 14 to 1. This setting ensures that a header received interrupt occurs on all 0x3C PID headers that occur on the LIN bus.

Error Handling

The ADuCM300 LIN interface is capable of handling multiple error events. The LINSTA register reflects the source of the error.

The following errors can produce an interrupt:

- Collision detection: in the event of a collision during transmission, the hardware automatically ceases transmission, and a maskable LININT1 interrupt is generated.
- Maximum negative edges: if the interface receives more than the maximum negative edges expected in a LIN frame, a maskable LININT1 interrupt occurs.
- PID parity error flag: a LIN PID has two parity bits. If the received PID does not correctly match the LIN2.2 parity scheme specification, this error is flagged. A new break symbol clears this flag.
- Framing error flag: in the event of a frame error, this error is flagged. A new break symbol clears this flag.
- Checksum error: if the received checksum does not match the calculated checksum, this error condition occurs.

LIN MEMORY MAPPED REGISTERS

Table 165. LIN Memory Mapped Registers (Base Address 0x40005000)

Offset	Name	Description	Access	Default
0x0000	LINCON	LIN control register	Read/write	0x1000
0x0004	LINIMSK	LIN interrupt mask register	Read/write	0x0000
0x0008	LINBR	LIN baud rate count register	Read/write	0x0326
0x000C	LINBRK	LIN break symbol count register	Read/write	0x0454
0x0014	LINSAMP	LIN sampling delay count register	Read/write	0x00
0x0018	LINFORCE	LIN force low register (forces LIN low while its value > 0)	Read/write	0x0000
0x001C	LINWUP	LIN wake-up register (minimum LIN bus low time needed to wake up the device)	Write	0x0007
0x0020	LINCNT	LIN byte count register (for the entire LIN frame)	Read/write	0x18
0x0024	LINTRDLY	LIN transceiver delay compensation register	Read/write	0x00
0x0028	LINSLP	LIN bus idle timeout register	Read/write	0x08
0x002C	LINLCNT	LIN last byte count register (number of bytes in the last response frame)	Read	0x0000
0x0030	LINSTA	LIN status register	Read/write	0x0000
0x0038	LINSTA1	LIN Status 1 register	Read	0x0000
0x003C	LINID	LIN PID register	Read	0x00
0x0040	LIND1	LIN Data Byte 1 for receive or transmit	Read/write	0x00
0x0044	LIND2	LIN Data Byte 2 for receive or transmit	Read/write	0x00
0x0048	LIND3	LIN Data Byte 3 for receive or transmit	Read/write	0x00
0x004C	LIND4	LIN Data Byte 4 for receive or transmit	Read/write	0x00
0x0050	LIND5	LIN Data Byte 5 for receive or transmit	Read/write	0x00
0x0054	LIND6	LIN Data Byte 6 for receive or transmit	Read/write	0x00
0x0058	LIND7	LIN Data Byte 7 for receive or transmit	Read/write	0x00
0x005C	LIND8	LIN Data Byte 8 for receive or transmit	Read/write	0x00
0x0060	LINFCS	LIN frame checksum register	Read	0x00
0x0064	LINCCS	LIN current calculated checksum register	Read	0xFF
0x0068	LINNAD	LIN node ID address register	Read/write	0xFF
0x006C	LININAD	LIN initial node ID register	Read/write	0x00
0x0070	LINFID	LIN initial function ID register	Read/write	0x0000
0x0074	LINVID	LIN variant ID register	Read/write	0x0000
0x0078	LINSUPID	LIN supplier ID register	Read/write	0x00
0x007C	LINSID	LIN service ID register	Read/write	0x00

LIN Control Register

Address: 0x40005000, Reset: 0x1000, Name: LINCON

Table 166. LINCON Register Bit Descriptions

Bits	Name	Description
15	TRDLY	Transceiver delay compensation. This bit, in association with the LINTRDLY register, is used to adjust the sampling point during transmit mode. The LINTRDLY register is used to compensate for transceiver transmit to receive delay time. 0: cleared to 0 to enable automatic tracking of transceiver delay time between transmit and receive (default). The LINTRDLY register is automatically updated on the beginning of every byte by measuring the falling edge transition of the start bit. 1: set to 1 to disable automatic delay compensation and enable manual programming of the LINTRDLY register. Software can write an appropriate value to compensate for transceiver delay time.
14	PIDRXINT	PID received interrupt (PID = 0x3C) enable. 0: cleared to 0 to disable header received interrupt for diagnostic frame with PID as 0x3C (default). 1: set to 1 to enable header received interrupt for diagnostic frame with PID as 0x3C.
13	WUPEN	Wake-up enable. 0: cleared to 0 to enable wake-up detection only when the core is in hibernate mode (default). 1: set to 1 to enable wake-up detection in hibernate mode, active mode, and system halt power mode. If wake-ups are not required to generate an interrupt in active mode or system halt mode, this bit must be cleared.
12	TXTIMEOUT	Transmit dominant timeout check. The transmit signal is checked by a running counter. If the transmit signal is held low for more than 100 ms, this counter logic forces the transmit signal high. 0: cleared to 0 to disable dominant timeout check on transmit signal. 1: set to 1 to enable dominant timeout check on transmit signal (default).
11	SYNC	Synchronization error check. The duration of the bits in the synchronization symbol in the LIN synchronization field are measured with respect to the duration of the start bit of the synchronization symbol. If the duration of successive bits in the synchronization symbol are either more than twice or less than half of the duration of the start bit, frame reception is aborted. 0: cleared to 0 to enable synchronization symbol error check. 1: set to 1 to disable synchronization symbol error check.
10	NAD	Node address for diagnostic (NAD) match. 0: cleared to 0 to enable NAD match check. Diagnostic frame is only recognized if NAD matches. 1: set to 1 to disable NAD match check. Diagnostic frame is always recognized.
9	COLL	Collision check. This bit is used to disable collision check using the sampled receive line and transmit data value. Collisions are checked only at the sampling instant in the response field if the direction bit (RXTXMODE), Register LINCON, Bit 1, is set. 0: cleared to 0 to enable collision check. This is the default state. 1: set to 1 to disable collision check.
8	SLEEPEN	Enable sleep counter. This bit enables the counter that monitors the LIN line for inactivity. The LINSLP register determines the bus idle timeout period. 0: cleared to 0 to disable sleep counter. 1: set to 1 to enable sleep counter.
7	CSCLR	Checksum clear. 1: this is a write only bit and when set to 1, clears the calculated checksum. This bit reads back a 0.
6	UARTEN	UART mode enable. This mode is used to transmit or receive data one byte at a time and is not LIN protocol compliant. LIND1 is the only register that participates in this mode. Register LINCON, Bit 1 must be cleared to allow receive of data bytes. 0: cleared to 0 for normal LIN protocol mode of operation. 1: set to 1 to allow transmission without receiving the frame header (UART mode).
5	BYPASSEN	LIN bypass enable. 0: cleared to 0 for LIN normal mode.
		1: set to 1 allows the user to take control of the LIN output of the LIN block (to perform LIN conformance tests).

Bits	Name	Description
4	SYNCTIM	Timing of synchronization symbol Bit 0 (not required in a single slave system). This bit ensures that if a second break is transmitted, the second break is recognized as such and not timed as part of the synchronization symbol. If the start symbol is more than the number of clock cycles dictated by this bit, the device assumes it is now receiving a break and continues to count the low cycle to verify if the break meets the minimum time required for a break, as defined in the LINBRK MMR. 0: cleared by user. The start bit of the synchronization symbol must be less than 8872 PCLK counts. 1: set by user. The start bit of the synchronization symbol must be less than 1209 PCLK counts.
3	LEN	Stop bit length. 0: if cleared, one stop bit is sent while transmitting. 1: if set, two stop bits are sent while transmitting.
2	CSCALC	Checksum calculation. This bit must be modified simultaneously with Register LINCON, Bit 0. 0: cleared to 0 to calculate an enhanced checksum, PID included. 1: set to 1 to calculate automatically a classic checksum, PID excluded.
1	RXTXMODE	Receive or transmit mode. 0: cleared to 0 when a break symbol is received or when transmit is complete. Also cleared when 0 is written to Register LINCON, Bit 0. 1: set to 1 to transmit data bytes after decoding the PID.
0	LINENABLE	LIN enable bit. 0: cleared to 0 by user code to disable the LIN interface or to reset the interface. 1: set to 1 by user code to enable the LIN interface. Clearing this bit resets all LIN interface registers to their default values, except for the following: Register LINSTA, Bits[4:0], Register LINSTA, Bits[13:11], Register LINCON, Bits[8:7], Register LINCON, Bits[13:12], and Register LINCON, Bit 15.

LIN Interrupt Mask Register

Address: 0x40005004, Reset: 0x0000, Name: LINIMSK

Table 167. LINIMSK Register Bit Descriptions

Bits	Name	Description
15	STOPSTART	Start and stop bit error interrupt mask. 0: cleared to 0 by user to enable the start and stop bit error detection. 1: set by user to disable the start and stop bit error detection.
14	SYNC	Synchronization field error interrupt mask. 0: cleared to 0 by user to enable the interrupt on synchronization field error detection on LININT1. 1: set by user to disable the interrupt synchronization field error detection on LININT1.
[13:12]	Reserved	Reserved. These bits must be set to 0.
11	UARTMODE	UART mode interrupt mask. 0: cleared to 0 by user to enable the interrupt in UART mode on LININT0. 1: set to 1 by user to disable the interrupt in UART mode on LININT0.
10	MAXBRK	Maximum break time interrupt mask. 0: cleared to 0 by the user to enable the maximum break time interrupt on LININT1. 1: set to 1 by the user to disable the maximum break time interrupt on LININT1.
9	MAXNEGEDGE	Maximum negative edges in the frame interrupt mask. An interrupt is generated if more than 57 falling edges are detected in a frame. 0: cleared to 0 by the user to enable the maximum negative edges in the frame interrupt on LININT1. 1: set to 1 by the user to disable the maximum negative edges in the frame interrupt on LININT1.
8	Wakeup	Wake-up interrupt mask. 0: cleared by the user to enable the wake-up interrupt on LININT2. 1: set to 1 by the user to disable the wake-up interrupt on LININT2.
7	Sleep	Sleep interrupt mask. 0: cleared by the user to enable the bus idle timeout sleep interrupt on LININT2. 1: set to 1 by the user to disable the bus idle timeout sleep interrupt on LININT2.

Bits	Name	Description
6	FRAMEERR	Frame error interrupt mask. An interrupt is generated if framing error is detected. 0: cleared to 0 by the user to enable the framing error interrupt on LININT1. 1: set to 1 by the user to disable the framing error interrupt on LININT1.
5	COLLDETECT	Collision detect interrupt mask. 0: cleared to 0 by the user to enable the collision detect interrupt on LININT1. 1: set to 1 by the user to disable the collision detect interrupt on LININT1.
4	CSERR	Checksum error detected interrupt mask. 0: cleared to 0 by the user to enable the checksum error detected interrupt on LININT1. 1: set to 1 by the user to disable the checksum error detected interrupt on LININT1.
3	PIDPARITY	PID parity error interrupt mask. 0: cleared to 0 by the user to enable the PID parity error interrupt on LININT1. 1: set to 1 by the user to disable the PID parity error interrupt on LININT1.
2	TXCOMP	Transmit complete interrupt mask. 0: cleared to 0 by the user to enable the transmit complete interrupt on LININT0. 1: set to 1 by the user to disable the transmit complete interrupt on LININT0.
1	FRAMERX	Frame received interrupt mask. An interrupt is generated when a complete frame is received. 0: cleared to 0 by the user to enable the frame received interrupt on LININT0. 1: set to 1 by the user to disable the frame received interrupt on LININT0.
0	HEADERRX	Header received interrupt mask. 0: cleared to 0 by the user to enable the header received interrupt on LININT0. 1: set to 1 by the user to disable the header received interrupt on LININT0. An interrupt is not generated for the master request frame (PID = 0x3C) if Register LINCON, Bit 14 is cleared.

LIN Baud Rate Count Register

Address: 0x40005008, Reset: 0x0326, Name: LINBR

Table 168. LINBR Register Bit Descriptions

Bits	Name	Description
[15:0]	BAUDRATE	Current baud rate value. User software must check this register to ensure that the baud rate is within expected limits. Baud rate = PCLK/LINBR.

LIN Break Symbol Count Register

Address: 0x4000500C, Reset: 0x0454, Name: LINBRK

Table 169. LINBRK Register Bit Descriptions

Bits	Name	Description
[15:0]	LINBRK	Contains the value after which the first break is considered valid. The value represents the time taken for 11 bits to be transmitted at 20 kbps (clock at 16 MHz). Break period = (LINBRK × 8 +7) × t_{PCLK}.

LIN Sampling Delay Count Register

Address: 0x40005014, Reset: 0x00, Name: LINSAMP

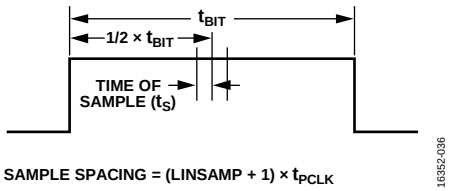

Figure 36. LIN Bit Sample Spacing

Table 170. LINSAMP Register Bit Descriptions

Bits	Name	Description
[7:0]	LINSAMP	Contains the count of clock cycles after which the LIN input line is sampled. Three such samples are taken, and the majority value of these three samples determines the value of the bit. Sample spacing = (LINSAMP + 1) × t_{PCLK}.

LIN Force Low Register

Address: 0x40005018, Reset: 0x0000, Name: LINFORCE

Table 171. LINFORCE Register Bit Descriptions

Bits	Name	Description
[15:0]	LINFORCE	Low period = LINFORCE × 8 × t_{PCLK}. If Register LINCON, Bit 12 is set and a transmit timeout has already occurred, a break must be detected to allow the LINFORCE register to force the LIN line low again.

LIN Wake-Up Register

Address: 0x4000501C, Reset: 0x0007, Name: LINWUP

Table 172. LINWUP Register Bit Descriptions

Bits	Name	Description
[15:0]	LINWAKEUP	Wake-up period > LINWUP × clock period (t_{LFOSC}). This is the minimum low period on the LIN line to guarantee a wake-up.

LIN Byte Count Register

Address 0x40005020, Reset: 0x18, Name: LINCNT

Table 173. LINCNT Register Bit Descriptions

Bits	Name	Description
[7:4]	Count	This field gives status information about the number of complete bytes sent and received. This number is updated at the end of the stop symbol of each byte.
[3:0]	NUMBYTES	Indicates the number of data bytes in the frame. Reset to 8 after the break is detected.

LIN Transceiver Delay Compensation Register

Address: 0x40005024, Reset: 0x00, Name: LINTRDLY

Table 174. LINTRDLY Register Bit Descriptions

Bits	Name	Description
[7:0]	TRDLY	This field can be manually configured by software or is filled automatically by hardware, depending on Register LINCON, Bit 15. If Register LINCON, Bit 15 is cleared, this register is automatically programmed by hardware to compensate for the transceiver delay time between transmit and receive (default). If Register LINCON, Bit 15 is set, manual programming of this register is required to compensate for the transmit to receive delay time. Software must write an appropriate value to compensate for the transceiver delay time. Delay adjustment = (LINTRDLY + 1) × t_{PCLK}. If the user is setting this value manually, the user needs to measure from transmit to receive and set the appropriate value as per the equation to reflect the delay.

LIN Bus Idle Timeout Register

Address: 0x40005028, Reset: 0x08, Name: LINSLP

Table 175. LINSLP Register Bit Descriptions

Bits	Name	Description
[4:0]	SLPVAL	This register defines the bus idle timeout period. When Register LINCON, Bit 8 is set, the hardware counts down from the timeout set by this register and generates a maskable interrupt, LININT2. The default value reflects a bus idle timeout of 4.096 sec. Bus idle period = LINSLP × 214 × 31.25 µs.

LIN Last Byte Count Register

Address: 0x4000502C, Reset: 0x0000, Name: LINLCNT

Table 176. LINLCNT Register Bit Descriptions

Bits	Name	Description
[3:0]	NUMLBYTES	This register indicates the number of data bytes in the previous LIN frame for both receive mode and transmit mode. Increment occurs for data bytes and CSUM byte. This register has a maximum value of 9, and the next counter iteration results in a value of 1. This register is not reset by hardware. Software in the LININT0 PID interrupt is expected to write any value to clear this register.

LIN Status Register

Address: 0x40005030, Reset: 0x0000, Name: LINSTA

Data written to the LINSTA register requires two low frequency oscillator clock periods before it is guaranteed to be latched.

Table 177. LINSTA Register Bit Descriptions

Bits	Name	Description
15	STOPSTART	Start and stop bit error. 0: cleared to 0 by a break detection or by writing 1 to this bit location. This bit is also cleared when 0 is written to Register LINCON, Bit 0. 1: set to 1 if the received signal start bit is sampled high or the stop bit is sampled low. This check is performed both while receiving or transmitting.
14	SYNCERR	Synchronization field error status. Generates a maskable interrupt (LININT1). 0: cleared to 0 by a break detect or by writing 1 to this bit location. This bit is also cleared when 0 is written to Register LINCON, Bit 0. 1: set to 1 when error is detected in the synchronization field. The synchronization bits in the synchronization field are timed by the duration of the synchronization field start bit. If the duration of successive bits are less than half or more than twice the duration of the start bit, the frame is aborted and this bit is set. This bit is also set if the overall duration of the synchronization field exceeds a maximum value.
13	COLL	Collisions bit value. This bit has meaning only if Register LINSTA, Bit 5 is set and indicates the status of the transmit bit when collision occurs. This bit is cleared by writing 1 to this bit location.
12	UARTDATA	UART data byte received status. Generates a maskable interrupt (LININT0). 0: cleared to 0 by reading the LIND1 register or by writing 1 to this bit location. 1: set to 1 when a byte is received into the LIND1 register in UART receive mode.
11	UARTTX	UART transmit complete status. Generates a maskable interrupt (LININT0). 0: cleared to 0 on a write to the LIND1 register or by writing 1 to this bit location. This bit is cleared when 0 is written to Register LINCON, Bit 0. 1: set to 1 when the contents of the LIND1 register are transmitted onto the LIN bus in UART transmit mode, and new data can be written to the LIND1 register.
10	BRKTIME	Break time maximum status. Generates a maskable interrupt (LININT1). 0: this bit is only cleared when the LIN interface is disabled, which is when Register LINCON, Bit 0 = 0. 1: set to 1 if a break symbol is longer than 22 bits. This maximum break time is based on the contents of the LINBRK register and can change if the contents of the LINBRK register change. Frame reception continues if this bit is set.
9	MAXNEGEDGE	Negative edge maximum error status. Generates a maskable interrupt (LININT1). 0: cleared to 0 if the number of negative edges allowed in a frame is not surpassed. This bit is also cleared when 0 is written to Register LINCON, Bit 0. 1: set to 1 if the number of negative edges is 57 or more. This bit is not required to be written by software and is cleared when a break is detected.
8	WAKESTA	Wake-up status. Generates a maskable interrupt (LININT2). 0: this bit is cleared when 0 is written to Register LINCON, Bit 0 or by writing 1 to this bit location. This bit must be cleared by software. 1: set to 1 when in sleep (power-down) mode if the LIN line is detected to be low for a longer time than programmed by LINWUP, followed by a rising edge.
7	SLEEPSTA	Sleep status. Generates a maskable interrupt (LININT2). 0: this bit is cleared when 0 is written to Register LINCON, Bit 0 or by writing 1 to this bit location. This bit must be cleared by software. 1: set to 1 if LIN bus remains at the same state for more than the value indicated by the LINSLP register.
6	FRMERR	Framing error status in receive mode. Generates a maskable interrupt (LININT1). 0: cleared to 0 when 0 is written to Register LINCON, Bit 0. This bit is cleared by writing 1 to this bit location. 1: set to 1 if a valid start and stop is not detected in receive mode, or if after receiving a sync, it is determined that the break was less than 11 bits long.
5	COLLDETECT	LIN collision detect status. Generates a maskable interrupt (LININT1). 0: cleared to 0 by break detect or when 0 is written to Register LINCON, Bit 0. 1: set to 1 automatically by the hardware if the device has stopped transmission due to a collision on the bus. Write 1 via the software to clear this bit. This bit is set only on a collision of data bits.

Bits	Name	Description
4	CSMATCH	Checksum match status. An error condition generates a maskable interrupt (LININT1) 0: cleared to 0 by break detect or by writing 1 to this bit location. 1: set to 1 when the checksum does not match. This is an error condition at the end of the frame, which generates an interrupt.
3	PIDPARITY	PID parity error status. Generates a maskable interrupt (LININT1). 0: cleared to 0 by a break detect or by writing 1 to this bit location. 1: set to 1 by hardware if the received PID byte does not correctly match the parity scheme for a PID, as described in the LIN2.2 specifications.
2	TXCOMP	Transmit complete. Generates a maskable interrupt (LININT0). 0: this bit is cleared by a break detect or by writing 1 to this bit location. 1: this bit is set when the last bit of the checksum byte is transmitted after transmitting all data bytes.
1	FRAMERX	Frame received status. Generates a maskable interrupt (LININT0). 0: cleared to 0 by break detect or by writing 1 to this bit location. 1: set to 1 if the full frame is received. If PID = 0x3C, NAD matches (or broadcast address). For all frames, checksum must match.
0	HEADERRX	Header received status. Generates a maskable interrupt (LININT0). 0: cleared to 0 by break detection or by writing 1 to this bit location. 1: set to 1 when a header is received. For PID = 0x3C, this status bit setting also depends on Register LINCON, Bit 14. PID parity check must also pass.

LIN Status 1 Register

Address: 0x40005038, Reset: 0x0000, Name: LINSTA1

Table 178. LINSTA1 Register Bit Descriptions

Bits	Name	Description
[15:1]	Reserved	Reserved.
0	FRMINCOMP	Incomplete frame received. 0: cleared to 0 by writing 1 to this bit location. 1: set to 1 when a break is detected and the expected number of bytes in the previous frame (Register LINCNT, Bits[3:0] + 1) is not the same as the actual number of bytes received (Register LINCNT, Bits[7:4]). This bit is not set if Register LINCNT, Bits[3:0] are equal to zero. The plus one is to account for the checksum byte. This status refers to the previous frame and is set for the first frame after enabling LIN. This register is superseded by the LINLCNT register. It is recommended to use the LINLCNT register to determine the completion status of the previous frame.

LIN PID Register

Address: 0x4000503C, Reset: 0x00, Name: LINID

Table 179. LINID Register Bit Descriptions

Bits	Name	Description
[15:0]	PIDRX	LIN PID received. Updated when the stop bit of the PID is received. Parity bits are not saved, but are checked in hardware.

LIN Data Byte Registers

LIN Data Byte 1 for Receive or Transmit

Address: 0x40005040, Reset: 0x00, Name: LIND1

Table 180. LIND1 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE1	Data Byte 1 for receive or transmit. Also used in UART mode. See Register LINCON, Bit 6 in Table 166.

LIN Data Byte 2 for Receive or Transmit

Address: 0x40005044, Reset: 0x00, Name: LIND2

Table 181. LIND2 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE2	Data Byte 2 for receive or transmit.

LIN Data Byte 3 for Receive or Transmit

Address: 0x40005048, Reset: 0x00, Name: LIND3

Table 182. LIND3 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE3	Data Byte 3 for receive or transmit.

LIN Data Byte 4 for Receive or Transmit

Address: 0x4000504C, Reset: 0x00, Name: LIND4

Table 183. LIND4 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE4	Data Byte 4 for receive or transmit.

LIN Data Byte 5 for Receive or Transmit

Address: 0x40005050, Reset: 0x00, Name: LIND5

Table 184. LIND5 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE5	Data Byte 5 for receive or transmit.

LIN Data Byte 6 for Receive or Transmit

Address: 0x40005054, Reset: 0x00, Name: LIND6

Table 185. LIND6 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE6	Data Byte 6 for receive or transmit.

LIN Data Byte 7 for Receive or Transmit

Address: 0x40005058, Reset: 0x00, Name: LIND7

Table 186. LIND7 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE7	Data Byte 7 for receive or transmit.

LIN Data Byte 8 for Receive or Transmit

Address: 0x4000505C, Reset: 0x00, Name: LIND8

Table 187. LIND8 Register Bit Descriptions

Bits	Name	Description
[7:0]	DATABYTE8	Data Byte 8 for receive or transmit.

LIN Frame Checksum Register

Address: 0x40005060, Reset: 0x00, Name: LINFCS

Table 188. LINFCS Register Bit Descriptions

Bits	Name	Description
[7:0]	LINFCS	While receiving a frame, a received checksum byte is saved into this register.

LIN Current Calculated Checksum Register

Address: 0x40005064, Reset: 0xFF, Name: LINCCS

Table 189. LINCCS Register Bit Descriptions

Bits	Name	Description
[7:0]	LINCCS	Current calculated checksum value.

LIN Node ID Address Register

Address: 0x40005068, Reset: 0xFF, Name: LINNAD

Table 190. LINNAD Register Bit Descriptions

Bits	Name	Description
[7:0]	NODEID	LIN node address register. For further information, refer to the LIN 2.2A specification.

LIN Initial Node ID Register

Address: 0x4000506C, Reset: 0x00, Name: LININAD

Table 191. LININAD Register Bit Descriptions

Bits	Name	Description
[7:0]	INITNODEID	LIN initial node address register. For further information, refer to the LIN 2.2A specification.

LIN Initial Function ID Register

Address: 0x40005070, Reset: 0x0000, Name: LINFID

Table 192. LINFID Register Bit Descriptions

Bits	Name	Description
[15:0]	FUNCID	LIN function ID register. For further information, refer to LIN 2.2A specification.

LIN Variant ID Register

Address: 0x40005074, Reset: 0x0000, Name: LINVID

Table 193. LINVID Register Bit Descriptions

Bits	Name	Description
[15:0]	VARID	LIN variant ID register. For further information, refer to the LIN 2.2A specification.

LIN Supplier ID Register

Address: 0x40005078, Reset: 0x00, Name: LINSUPID

Table 194. LINSUPID Register Bit Descriptions

Bits	Name	Description
[7:0]	SUPID	LIN supplier ID register. For further information, refer to the LIN 2.2A specification.

LIN Service ID Register

Address: 0x4000507C, Reset: 0x00, Name: LINSID

Table 195. LINSID Register Bit Descriptions

Bits	Name	Description
[7:0]	SERVID	LIN service ID register. For further information, refer to the LIN 2.2A specification.

DEVICE IDENTIFICATION

For traceability, device identification is available at power-up. This traceability information is contained across five registers: R4, FEEDATL, ChipID, System Serial ID 0, and System Serial ID 1.

R4

After kernel execution, this 32-bit register, R4, holds the assembly lot ID. This information must be saved to a known SRAM location before user code is executed. This register belongs to the Cortex-M3 processor.

FEEDATL REGISTER

Address: 0x40018010, Reset: 0xF300130F, Name: FEEDATL

After kernel execution, the FEEDATL MMR contains information that can be used to identify the ADuCM300. This information must be saved to a known SRAM location before user code is executed.

Table 196. FEEDATL Register Bit Descriptions

Bits	Description
[31:16]	Identity of the ADuCM300. Confirm the device identification. ADuCM300: 0xF300.
[15:0]	Silicon mask revision ID, kernel revision ID, and kernel minor revision number. Same as Register SYSSER1, Bits[31:16], 0xXXXX

Chip ID REGISTER

The chip identification register is a 16-bit field that identifies the implemented cortex low power platform and the silicon revision.

Address: 0x40002024, Reset: 0x53, Name: CHIPID

Table 197. ChipID Register

Offset	Name	Description	Access	Default
0x0000	CHIPID	Chip identification register	Read	0x0053

Table 198. ChipID Register Bit Descriptions

Bits	Description
[15:4]	Digital die hardware identification, 0x5 = ADuCM300 Cortex-M3 device.
[3:0]	Digital die hardware revision. 0x3 = ADuCM300/ADuCM330/ADuCM331

SYSTEM SERIAL ID REGISTERS

Table 199. System Serial ID Memory Mapped Registers (Base Address 0x000207F0)

Offset	Name	Description	Access	Default
0x0000	SYSSER0	System Serial ID 0	Read	0xXXXXXXXX
0x0004	SYSSER1	System Serial ID 1	Read	0xXXXXXXXX

System Serial ID 0

Address: 0x000207F0, Reset: 0xXXXXXXXX, Name: SYSSER0

A 32-bit location that holds the value of the original manufacturing lot number from which this specific ADuCM300 device was manufactured (bottom die only). Used in conjunction with SYSSER1, this lot number allows the full manufacturing history of this device to be traced (bottom die only).

Table 200. SYSSER0 Register Bit Descriptions

Bits	Description
[31:27]	Wafer number. The five bits read from this location give the wafer number (1 to 24) from the wafer fabrication lot ID (from which this device originated).
[26:22]	Wafer lot fabrication plant. The five bits read from this location reflect the manufacturing plant associated with this wafer lot.
[21:16]	Wafer lot fabrication ID. The six bits read from this location form part of the wafer lot fabrication.
[15:0]	Wafer lot fabrication ID. These 16 LSBs hold a 16-bit number to be interpreted as the wafer fabrication lot ID number.

System Serial ID 1

Address: 0x00207F4, Reset: 0xXXXXXXXX, Name: SYSSER1

A 32-bit location that holds the values of the device ID number, silicon mask revision number, and kernel revision number of the bottom die only, as detailed in Table 201.

Table 201. SYSSER1 Register Bit Descriptions

Bits	Description
[31:28]	Silicon mask revision ID. The four bits read from this nibble reflect the silicon mask ID number. The value range for the ADuCM300 is interpreted as 0x41 to 0x4F, reflecting the ASCII A to O characters.
[27:20]	Kernel revision ID. This byte contains the hexadecimal number, which is interpreted as an ASCII character indicating the revision of the kernel firmware embedded in the on-chip Flash/EE memory. For example, reading 0x31 from this byte is interpreted as 1, indicating a Revision 1 kernel is on chip.
[19:16]	These bits refer to the kernel minor revision number of the device.
[15:0]	Device ID. These 16 LSBs hold a 16-bit number that is interpreted as the device ID number. When used in conjunction with the value in SYSSER0 (the manufacturing lot ID), this number is a unique identifier for the device.

MMR LISTING

Table 202. Memory Map Summary

Base Address	Peripheral
0x000207F0	System serial ID
0x40000000	General-Purpose Timer 0
0x40002000	SRAM, clock control, and hardware identification
0x40002400	Interrupt detection, power mode, and reset
0x40002500	Wake-up timer
0x40002580	Watchdog timer
0x40003000	High voltage control interface
0x40004000	SPI
0x40005000	LIN
0x40005C00	High frequency oscillator calibration
0x40006000	GPIO
0x40008824	Reference control
0x40009C00	Low frequency oscillator calibration
0x40018000	Flash controller
0x40030000	ADC
0xE000E004	Cortex-M3 and NVIC