ANALOG DEVICES

AN-1111
APPLICATION NOTE

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

Options for Minimizing Power Consumption When Using the ADuCM360/ADuCM361

By Mike Looney

INTRODUCTION

The ADuCM360 is a 32-bit ARM® Cortex™-M3-based microcontroller that integrates dual 24-bit sigma delta (Σ-Δ) analog-to-digital converters (ADCs), each with a fully programmable instrumentation amplifier on the front end. The ADuCM361 contains all of the features of the ADuCM360 except that it has a single 24-bit sigma delta ADC (ADC1). This application note describes many of the operational modes and clock options for these devices and details the power savings for each configuration.

These microcontrollers target a wide range of applications including industrial control and instrumentation applications. In many of the target applications, reducing power consumption in the application is of the utmost importance. For example, for battery-powered applications, the lifetime of the battery can be extended by using the many operating modes and clock options on the ADuCM360/ADuCM361. In addition, in 4 mA to 20 mA loop-based applications where ADC performance is important, the ADuCM360/ADuCM361 ensure that the overall power consumption of the sensor module remains below 3.2 mA.

For details regarding the specifications and operation of the ADuCM360/ADuCM361, refer to the latest datasheet and the UG-367 user guide.

By maximizing the use of clock and power mode options, the average I_{DD} consumed by the ADuCM360/ADuCM361 can be reduced to just 1 mA. This figure is achieved despite both the ADCs and the PGAs on the front end being enabled, the ARM Cortex-M3 processor being set to active mode, and the SPI buses and all timers being enabled.

All I_{DD} measurements in this document are typical values measured at ambient temperature (25°C) using a supply voltage of AV_{DD} = IOVDD = 3.0 V.

TABLE OF CONTENTS

REVISION HISTORY

/ ./12—Revision 0: Initial Version

CLOCK CONTROL REGISTERS

The ADuCM360/ADuCM361 contain four main clock control registers: CLKCON0, CLKCON1, CLKDIS, and CLKSYSDIV. Figure 1 provides an overview of the ADuCM360/ADuCM361 clock structure.

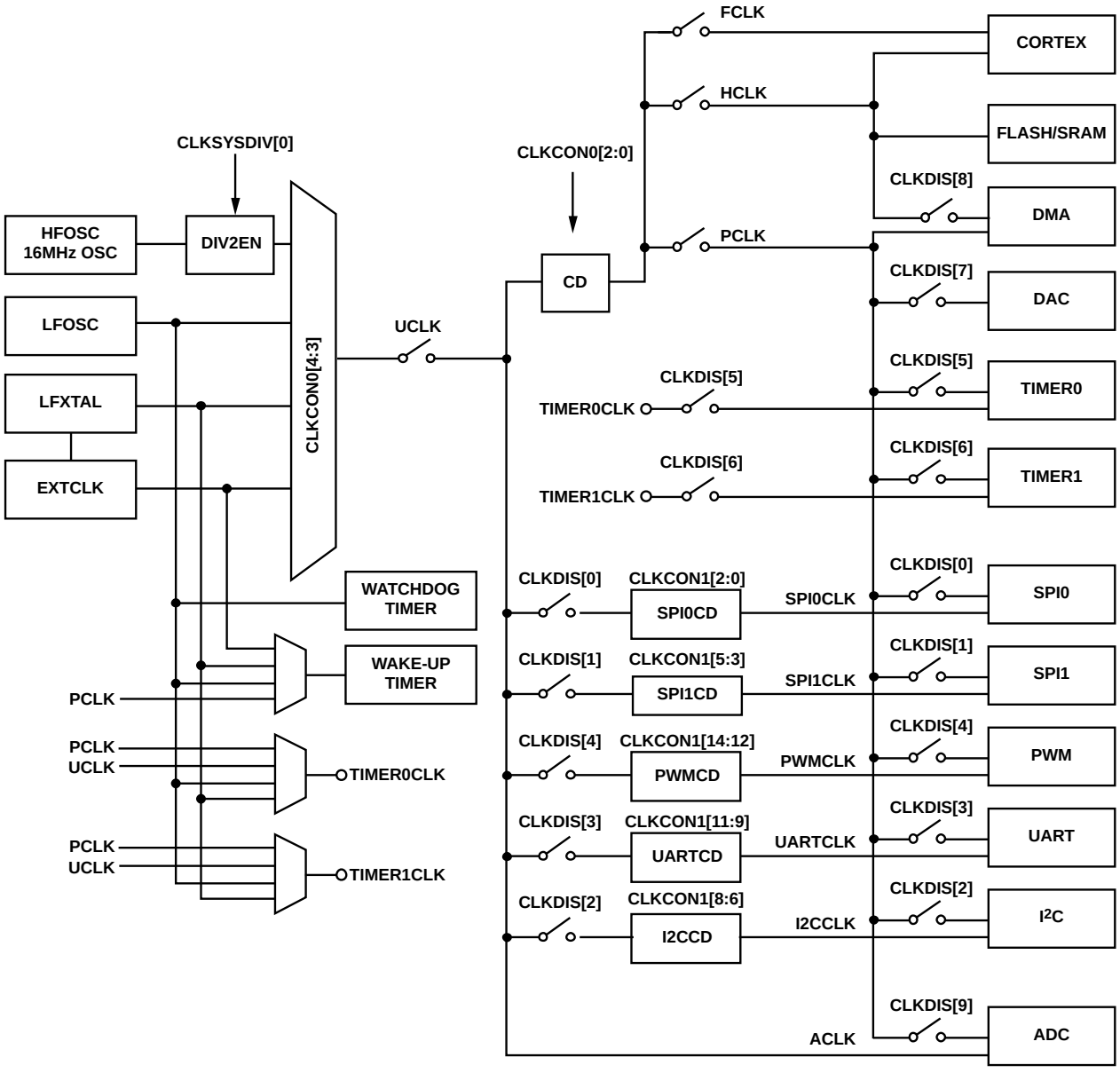

Figure 1. Clock Structure for the ADuCM360 and ADuCM361

CLKCON0

UCLK serves as the main system clock for both the ADuCM360 and the ADuCM361. The CLKCON0[2:0] bits in Register CLKCON0 (Address 0x40002000) select the clock divide value for UCLK. The clock divide setting is important because a lower system clock setting reduces power consumption.

Figure 2 shows the I_{DD} of the ADuCM360/ADuCM361 for different UCLK rates. In Figure 2, the processor is enabled but the ADCs are turned off.

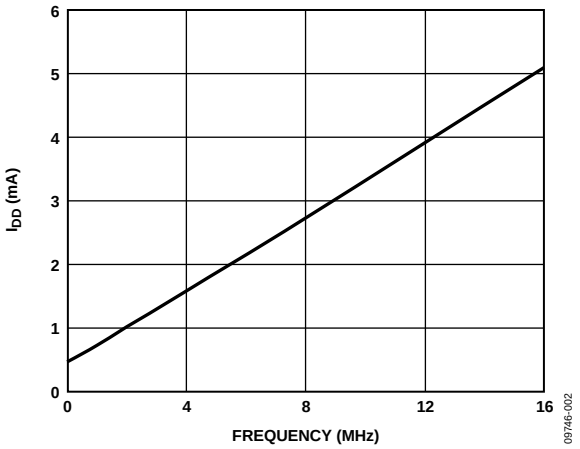

Figure 2. Total I_{DD} Using a 3.0 V Supply with the Processor Running at Different Frequencies

The CLKCON0[4:3] bits select the source of UCLK. The options include the following:

- Internal 16 MHz oscillator (default), HFOSC
- Internal 32 kHz oscillator, LFOSC
- External 32 kHz oscillator, LFXTAL
- External clock from P1.0, EXTCLK

The 16 MHz internal oscillator (HFOSC), by default, uses 170 μA.

CLKSYSDIV

The CLKSYSDIV register (Address 0x40002444) enables and disables a divide-by-two (DIV2EN) option on the output of the 16 MHz oscillator.

When CLKSYSDIV = 0x1, the system clock (UCLK) becomes 8 MHz; therefore, the entire chip is clocked from an 8 MHz clock source instead of 16 MHz. This has the effect of not only halving the dynamic current shown in Figure 3 but also reducing the background (static) current by 90 μA typically.

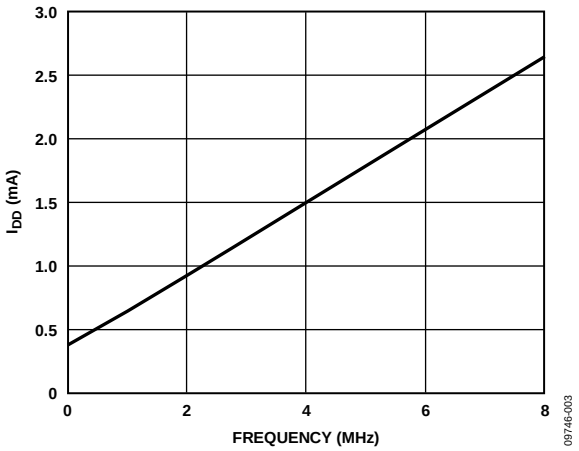

Figure 3. Total I_{DD} Using a 3.0 V Supply with the Processor Running, CLKSYSDIV = 0x1

Table 1. CLKSYSDIV Register Bit Descriptions

Bit	Name	Description
[7:1]	Reserved	These bits are reserved and are cleared to 0.
0	DIV2EN	Divide-by-two enable bit. By default, this bit is 1, meaning that the system clock is 8 MHz. Enable this bit in low power systems. 1: enable the system clock divider; the system clock is 8 MHz. 0: disable the system clock divider; the system clock is 16 MHz.

CLKDIS

The CLKDIS register (Address 0x4000202C) enables and disables the system clock to 10 different peripherals as shown in Figure 1. By default, all CLKDIS bits are set to 1 except Bit 9, the ADC system clock enable bit. This disables the system clock to 9 of the 10 of these peripherals after a reset. To use any one of these 10 peripherals, the user must clear the appropriate bit in the CLKDIS register to enable the peripheral system clock.

If an application does not use any of the 10 peripherals, to minimize power, set the CLKDIS register bit for each unused peripheral to 1.

Table 2 lists the I_{DD} savings by disabling the clock to each peripheral by setting CLKCON1 = 0x00, assuming that CLKSYSDIV = 0x01 (system clock = 8 MHz). In most cases, by default, the peripheral is clocked by the system clock although the peripheral remains inactive. For more information about the CLKCON1 register, see the CLKCON1 section.

Table 2. CLKDIS Register Bit Descriptions

Bit	Name	Description	I_{DD} Reduction
[15:10]	Reserved	These bits are reserved and cleared to 0	
9	DISADCCLK	1: disable ADC system clock 0: enable ADC system clock	75 µA reduction when set to 1. Note that ADC0 and ADC1 are in idle mode by default, ADCxMDE register = 0x0003.
8	DISDMACLK	1: disable DMA system clock 0: enable DMA system clock	40 µA reduction when set to 1. Note that by default, all DMA channels are disabled.
7	DISDACCLK	1: disable DAC system clock 0: enable DAC system clock	12 µA reduction when set to 1. Note that the DAC, by default, is in power-down mode, DACCON register = 0x200.
6	DIST1CLK	1: disable Timer1 system clock 0: enable Timer1 system clock	20 µA reduction when set to 1.
5	DIST0CLK	1: disable Timer0 system clock 0: enable Timer0 system clock	15 µA reduction when set to 1.
4	DISPWMCLK	1: disable PWM system clock 0: enable PWM system clock	95 µA reduction when set to 1.
3	DISUARTCLK	1: disable UART clock 0: enable UART system clock	135 µA reduction when set to 1.
2	DISI2CCLK	1: disable I^2C system clock 0: enable I^2C system clock	70 µA reduction when set to 1.
1	DISSPI1CLK	1: disable SPI1 system clock 0: enable SPI1 system clock	80 µA reduction when set to 1.
0	DISSPI0CLK	1: disable SPI0 system clock 0: enable SPI0 system clock	85 µA reduction when set to 1.

CLKCON1

The CLKCON1 register (Address 0x40002004) scales the clock to the main communications peripherals: SPI0, SPI1, UART, I²C, and PWM. By default, the CLKCON1 register is 0x0000.

To minimize I_{DD} when any of the SPI, I²C, UART, or PWM peripherals are unused in an application, disable the clock to that peripheral and set the clock frequency to its minimum value. For example, if the PWM is not used, set CLKCON1, Bits[14:12] to 111. Note that the peripheral clock must be equal to or faster than the processor clock speed. If the peripheral clock is slower than the processor clock speed, the peripheral is disabled.

Table 3. CLKCON1 Register Bit Descriptions[1]

Bit	Name	Description
15	Reserved	
[14:12]	PWMCD	Clock divide bits for PWM system clock 000: UCLK/1 = 16 MHz 001: UCLK/2 = 8 MHz 010: UCLK/4 = 4 MHz 011: UCLK/8 = 2 MHz 100: UCLK/16 = 1 MHz 101: UCLK/32 = 500 kHz 110: UCLK/64 = 250 kHz 111: UCLK/128 = 125 kHz
[11:9]	UARTCD	Clock divide bits for UART system clock 000: UCLK/1 = 16 MHz 001: UCLK/2 = 8 MHz 010: UCLK/4 = 4 MHz 011: UCLK/8 = 2 MHz 100: UCLK/16 = 1 MHz 101: UCLK/32 = 500 kHz 110: UCLK/64 = 250 kHz 111: UCLK/128 = 125 kHz
[8:6]	I2CCD	Clock divide bits for I²C system clock 000: UCLK/1 = 16 MHz 001: UCLK/2 = 8 MHz (the minimum value to support a 400 kHz I²C baud rate) 010: UCLK/4 = 4 MHz 011: UCLK/8 = 2 MHz (the minimum value to support a 100 kHz I²C baud rate) 100: UCLK/16 = 1 MHz 101: UCLK/32 = 500 kHz 110: UCLK/64 = 250 kHz 111: UCLK/128 = 125 kHz
[5:3]	SPI1CD	Clock divide bits for SPI1 system clock 000: UCLK/1 = 16 MHz 001: UCLK/2 = 8 MHz 010: UCLK/4 = 4 MHz 011: UCLK/8 = 2 MHz 100: UCLK/16 = 1 MHz 101: UCLK/32 = 500 kHz 110: UCLK/64 = 250 kHz 111: UCLK/128 = 125 kHz
[2:0]	SPI0CD	Clock divide bits for SPI0 system clock 000: UCLK/1 = 16 MHz 001: UCLK/2 = 8 MHz 010: UCLK/4 = 4 MHz 011: UCLK/8 = 2 MHz 100: UCLK/16 = 1 MHz 101: UCLK/32 = 500 kHz 110: UCLK/64 = 250 kHz 111: UCLK/128 = 125 kHz

[1] Calculations are for UCLK = 16 MHz with CLKSYSDIV[0] = 0; an additional divide-by-two is required when CLKSYSDIV[0] is set to 1.

UART

To minimize the current drawn by the UART, use the lowest possible clock setting that allows the application to comply with the required UART baud rate setting. Changing the clock frequency to the UART likewise requires recalculating the correct values that are entered into the COMDIV register, which is the register that controls the baud rate.

To calculate the baud rate, use the following equation:

$$Baud\ Rate = UARTCLK \div (2 \times 16 \times COMDIV) \div (M + N \div 2048)$$

where:
COMDIV = 1 to 65,536.
M = 1 to 3.
N = 0 to 2047.

UARTCLK is the divided system clock to the UART set up by the CLKSYSDIV and CLKCON1 registers. Figure 4 shows the possible I_{DD} savings when the UART clock is reduced.

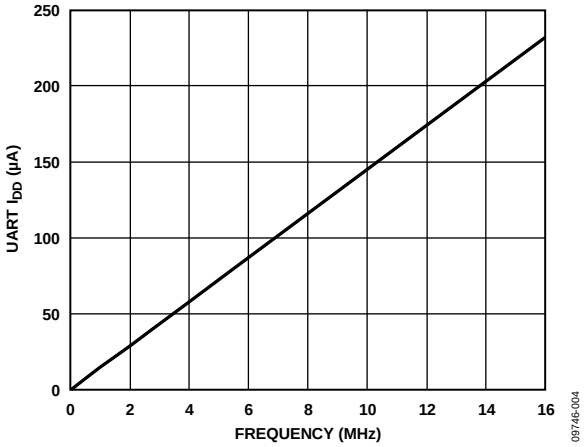

Figure 4. UART I_{DD} vs. UART Clock Frequency

SPI0/SPI1

To minimize the current drawn by the SPI block, use the lowest possible clock setting that allows the application to comply with the required SPI clock rate. Figure 5 shows the I_{DD} savings from each SPI block when a reduced clock is selected.

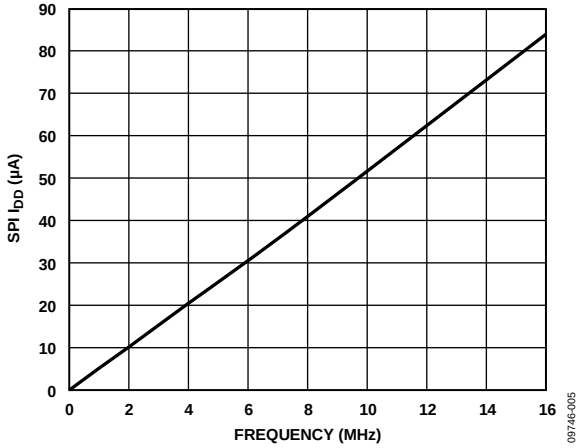

Figure 5. I_{DD} vs. SPI Clock Frequency for Each SPI Port

Master Mode

Changing the clock frequency to the SPI while in SPI master mode requires recalculating the correct values to be entered into the SPIxDIV register (where x is 0 for SPI0 and 1 for SPI1), which is the register that controls the SPI clock rate. Calculate the baud rate as follows:

$$SPI\ Clock\ Rate = SPI\ Clock/(2 \times (1 + SPIxDIV))$$

where *SPI Clock* is the divided system clock to the SPI set up by the CLKSYSDIV and CLKCON1 registers.

Slave Mode

In slave mode, the master on the SPI bus controls the baud rate. However, the internal ADuCM360/ADuCM361 SPI clock rate, set by the CLKCON1 register, must be at least four times faster than the master SPI output clock (host clock).

I²C

To minimize the current drawn by the I²C block, use the lowest possible clock setting that allows the application to comply with the required I²C clock rate. Figure 6 shows the possible I_{DD} savings when the I²C clock is reduced. Note that these figures do not include the I_{DD} from the external pull-up resistors on the SDA and SCL pins.

The minimum value to support a 400 kHz I²C baud rate is an I²C system clock of 8 MHz. Whereas, the minimum value to support a 100 kHz I²C baud rate is an I²C system clock of 2 MHz.

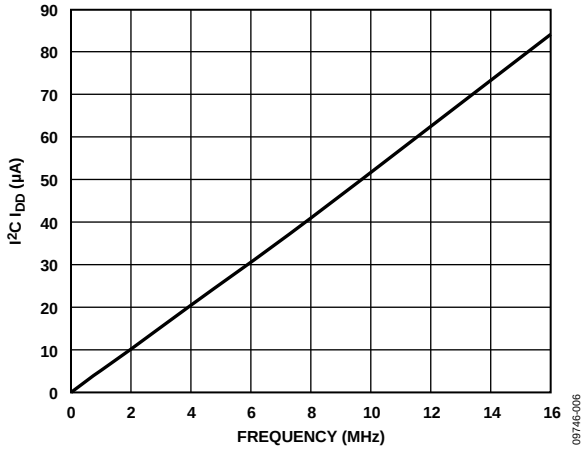

Figure 6. I_{DD} vs. I²C System Clock Frequency

Master Mode

Changing the clock frequency to the I²C block while in I²C master mode requires recalculating the correct values for entering data to the I2CDIV register, which is the register that controls the I²C clock rate. The I2CDIV is a 16-bit register containing two 8-bit values, high and low. This is set up according to the formula:

$$f_{I2CSCL} = f_{PERIPH}/(Low + High + 3)$$

where:
f_{PERIPH} = is the I²C peripheral clock.
f_{PERIPH} = f_{UCLK}/(CLKSYSDIV × I2CCD) where UCLK is the system clock, 16 MHz;. CLKSYSDIV is 1 or 2, depending on the CLKSYSDIV[0] bit setting; and I2CCD is the clock divide value

and is set by the CLKCON1[8:6] bits from 1 to 7.
Low = the low period of the clock, I2CDIV[7:0] = (*REQD_LOW_TIME*/*UCLK_PERIOD*) − 1.
High = the high period of the clock, I2CDIV[15:8] = (*REQD_HIGH_TIME*/*UCLK_PERIOD*) − 2.

Thus, for 100 kHz operation with an I²C peripheral clock of 16 MHz, the low and high bit values are as follows:

Low = 0x4F
High = 0x4E

For 400 kHz operation, the low and high bit values are as follows:

Low = 0x13
High = 0x12

Slave Mode

In slave mode, the master on the I²C bus controls the baud rate.

PWM

To minimize the current drawn by the PWM block, use the lowest possible clock setting that allows the application to meet the required PWM duty cycle and output frequency. Figure 7 shows the possible I_{DD} savings when reducing the PWM clock.

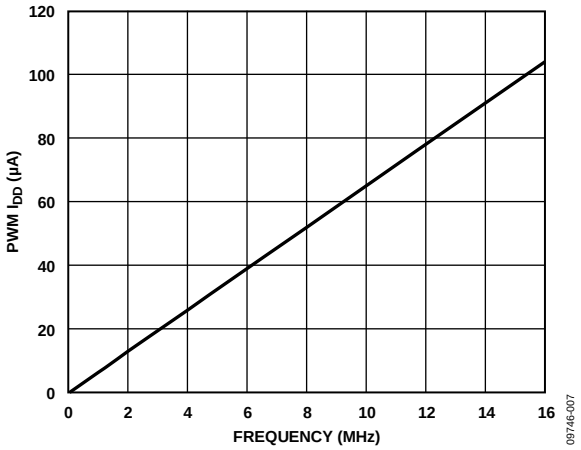

Figure 7. I_{DD} vs. PWM Clock Frequency

REDUCING ANALOG I_{DD}

Table 4 lists some of the options when configuring ADC0 and ADC1 on the ADuCM360/ADuCM361, and the associated I_{DD} values for each option. As Table 4 shows, the analog I_{DD} can be minimized by carefully configuring the ADCs and the DAC.

It is important to consider the following information when reducing the analog I_{DD}:

- The ADC update rate selected by the ADCxFLT registers does not affect the I_{DD} consumption. Regardless of the ADC filter update rate, the ADC modulator always works with a 500 kHz clock source.
- When the PGA is enabled and the gain is greater than or equal to 2, the ADC input buffers are not required; therefore, Register ADCxCON, Bits[17:14] can be set to 1111. This setting saves 70 µA per ADC.
- When the PGA is set to a gain greater than or equal to 32, an additional 60 µA is consumed by the PGA compared to gains of less than 32.
- When the PGA is disabled (gain = 1), enable the ADC input buffers, unless an external buffer is provided.
- The external reference input buffers consume 60 µA each. If the external reference negative voltage is connected to AGND, the negative input buffer can be bypassed and powered by setting Register ADCxCFG, Bits[1:0] = 11.
- Register IEXCCON, Bit 2 = 0 disables the Excitation Current Source 0. Similarly, Register IEXCCON, Bit 5 = 0 disables Excitation Current Source 1. When only one excitation current is used, disable the other one to save power.

Table 4. Breakdown of Analog Peripherals I_{DD}

Peripheral Name	ADC0	ADC1	Common to Both ADCs	Other
Modulator	70 µA	70 µA		
Gain = 2, 4, 8, or 16 (PGA Total)	130 µA	130 µA		
Gain = 32, 64, or 128 (PGA Total)	190 µA	190 µA		
ADC Positive Input Buffer	35 µA	35 µA		
ADC Negative Input Buffer	35 µA	35 µA		
External Reference Buffer				
Positive			60 µA	
Negative			60 µA	
Excitation Current (Excluding Output Current)				
Source 0			25 µA	
Source 1			25 µA	
DAC				50 µA

POWER-DOWN MODES

The ADuCM360/ADuCM361 provide five power-down levels. When a user is deciding on which power-down mode best suits an application, there is a trade-off among the I_{DD} savings, the wake-up time, and which of the peripherals needs to be active.

In Mode 1, Mode 2, and Mode 3, it is possible for peripherals to continue operating by using the DMA operation while the CPU is powered down. To achieve this, either ADC1 or ADC0 must be enabled for DMA operation. Use a DMA complete interrupt to awaken the device from Mode 1, Mode 2, or Mode 3.

The I_{DD} figures in the following sections assume that the ADC DMA mode is not enabled.

MODE 1: MCUHALT MODE

In Mode 1, HCLK is off and the ARM Cortex-M3 processor is in sleep mode. The wake-up time is three to five times FCLK, where FCLK is the clock selected by the CLKCON0[2:0] bits.

Calculate the expected I_{DD} in Mode 1 as follows:

When Register CLKSYSDIV = 0x0

$$I_{DD}\ [\mu A] = 50 \times FCLK + 355$$

When Register CLKSYSDIV = 0x1

$$I_{DD}\ [\mu A] = 50 \times FCLK + 435$$

MODE 2: PERHALT MODE

In Mode 2, PCLK is off and the ARM Cortex-M3 processor is in sleep mode. The wake-up time is three to five times FCLK, where FCLK is the clock selected by the CLKCON0[2:0] bits.

Calculate the expected I_{DD} in Mode 2 as follows:

When Register CLKSYSDIV = 0x0

$$I_{DD}\ [\mu A] = 60 \times FCLK + 345$$

When Register CLKSYSDIV = 0x1

$$I_{DD}\ [\mu A] = 60 \times FCLK + 425$$

The I_{DD} savings between Mode 1 and Mode 2 are minimal.

MODE 3: SYSHALT MODE

In Mode 3, HCLK, ACLK, and PCLK are off and the ARM Cortex-M3 processor is in sleep mode. The wake-up time is three to five times FCLK, where FCLK is the clock selected by the CLKCON0[2:0] bits.

Calculate the expected I_{DD} in Mode 3 as follows:

When Register CLKSYSDIV = 0x0

$$I_{DD}\ [\mu A] = 16 \times FCLK + 345$$

When Register CLKSYSDIV = 0x1

$$I_{DD}\ [\mu A] = 16 \times FCLK + 420$$

The I_{DD} savings between Mode 3 and Mode 1 or between Mode 3 and Mode 2 are minimal at FCLK rates ≤ 1 MHz.

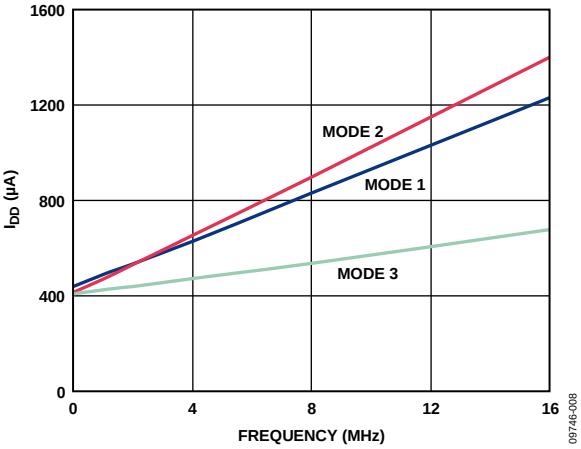

Figure 8. Power-Down I_{DD} for Mode 1, Mode 2, and Mode 3 vs. FCLK, CLKSYSDIV = 0x0

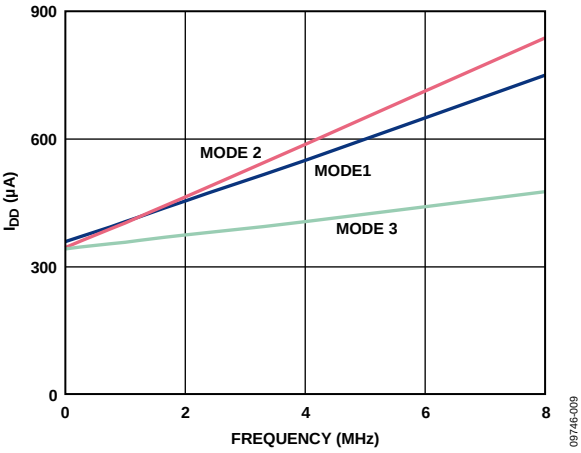

Figure 9. Power-Down I_{DD} for Mode 1, Mode 2, and Mode 3 vs. FCLK, CLKSYSDIV = 0x1

MODE 4 AND MODE 5: TOTALHALT AND HIBERNATE MODES

In Mode 4 and Mode 5, HCLK, ACLK, and PCLK are off and the ARM Cortex-M3 processor is in DEEPSLEEP mode. The wake-up time is ~30.8 µs. The typical I_{DD} in these modes is 2 µA to 4 µA.

NOTES

NOTES

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

AN09746-0-10/12(0)