ADuCM310 Hardware Reference Manual

UG-549

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

How to Set Up and Use the ADuCM310

SCOPE

This reference manual provides a detailed description of the ADuCM310 functionality and features.

Full specifications on the ADuCM310 are available in the product data sheet, which must be consulted in conjunction with this reference manual when using the device.

FUNCTIONAL BLOCK DIAGRAM

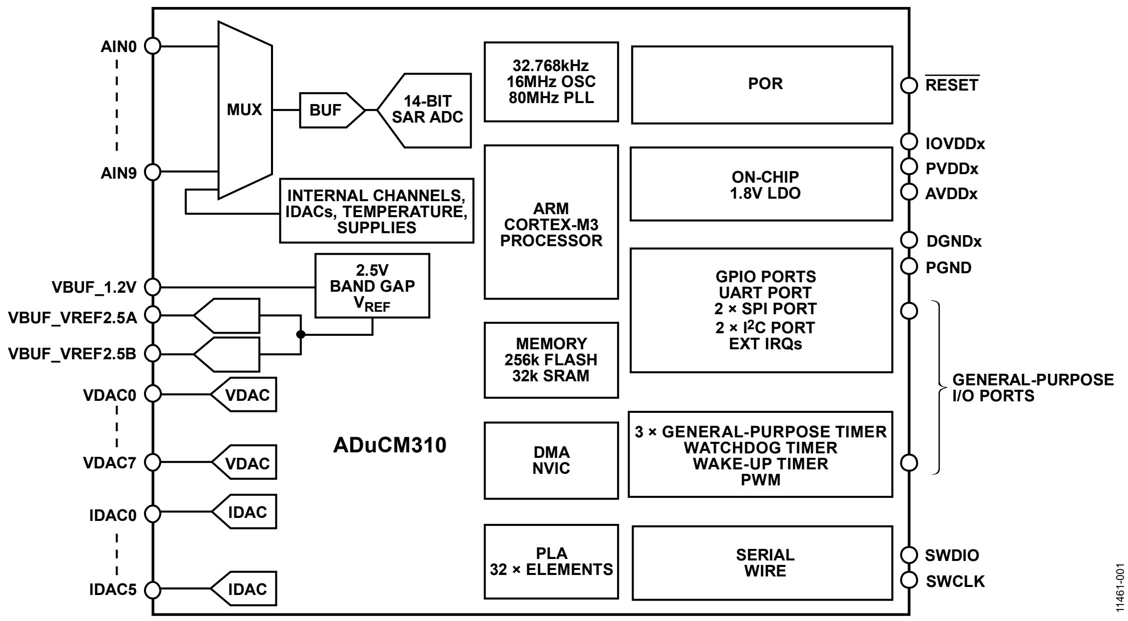


Figure 1.

TABLE OF CONTENTS

REVISION HISTORY

6/2019—Rev. A to Rev. B
Added Table 44 Title 48
Changes to Table 201 141

3/2016—Rev. 0 to Rev. A
Changes to Table 18 30
Deleted Endnote 1, Table 74; Renumbered Sequentially 64
Changes to Protection, Integrity Section 78
Added ECC Error Handling Section 81
Added ECC Error During Read Section and ECC Error During Execution of Sign Command 82
Changes to Table 104 86
Changes to Table 105 87
Added ECC Enable/Disable, Error Response Register Section and Table 121; Renumbered Sequentially 93
Added Flash 0 ECC Error Address Register Section, Table 122, Flash 1 ECC Error Address Register Section, and Table 123 94
Change to Table 130 99

5/2015—Revision 0: Initial Version

USING THE ADuCM310 HARDWARE REFERENCE MANUAL

NUMBER NOTATIONS

Table 1.

Notation	Description
Bit N	Bits are numbered in little endian format, that is, the least significant bit of a number is referred to as Bit 0.
V[x:y]	Bit field representation covering Bit x to Bit y of a value or a field (V).
0xNN	Hexadecimal (Base 16) numbers are preceded by the prefix 0x.
0bNN	Binary (Base 2) numbers are preceded by the prefix 0b.
NN	Decimal (Base 10) numbers are represented using no additional prefixes or suffixes.

REGISTER ACCESS CONVENTIONS

Table 2.

Mode	Description
RW	Memory location has read and write access.
RC	Memory location is cleared after reading it.
R	Memory location is read access only. A read always returns 0, unless otherwise specified.
W	Memory location is write access only.

MMR bits that are not documented are reserved. Reserved bits must not be changed by the user.

ACRONYMS AND ABBREVIATIONS

Table 3.

Acronym/Abbreviation	Description
ADC	Analog-to-digital converter
DMA	Direct memory access
GPIO	General-purpose input and output
LSB	Least significant byte/bit
MMR	Memory mapped register
MSB	Most significant byte/bit
NMI	Nonmaskable interrupt
NVIC	Nested vectored interrupt controller
Rx	Receive
SAR	Successive approximation register
SOA	Semiconductor optical amplifier
SPI	Serial peripheral interface
SWD	Sync word detect/serial wire debug
Tx	Transmit
UART	Universal asynchronous transmitter
WDT	Watchdog timer
WUT	Wake-up timer

INTRODUCTION TO THE ADuCM310

The ADuCM310 is a triple-die stack, system on-chip designed for diagnostic control on tunable laser optical module application. The ADuCM310 features a 16-bit (14-bit no missing codes), multichannel, successive approximation register (SAR) ADC; an ARM Cortex™-M3 processor; eight voltage DACs; six current output DACs; and Flash/EE memory packaged in a 6 mm × 6 mm, 112-ball BGA package.

The bottom die in the stack supports the bulk of the low voltage analog circuitry and is largest of the three die. It contains the ADC, VDAC, main IDAC circuits, as well as other analog support circuits like the low drift, precision 2.5 V voltage reference source.

The middle die in the stack supports the bulk of the digital circuitry including the ARM Cortex-M3 processor, the flash and SRAM blocks, and all of the digital communication peripherals. Also provided on this die are the clock sources for the whole chip. A 16 MHz internal oscillator is the source of an internal phase-locked loop (PLL) that outputs an 80 MHz system clock.

The top die, which is the smallest die, is developed on a high voltage process and supports the −5 V and +5 V VDAC outputs. It also implements the SOA IDAC current sink circuit, which allows the external SOA diode to be pulled to a −3.0 V level to implement the fast shutdown of the laser module.

Regarding the individual blocks, the ADC is capable of operating at conversion rates of up to 800 kSPS. There are 10 external inputs to the ADC, and these can be configured as single-ended or differential. Several internal channels are available, including supply monitor channels, an on-chip temperature sensor, and internal voltage reference monitor.

The voltage DACs are 12-bit string DACs with output buffers capable of sourcing between 10 mA and 50 mA, and all are capable of driving 10 nF capacitive loads.

The low drift current DACs have 14-bit resolution and have varied full-scale output ranges from 0 mA to 20 mA to 0 mA to 250 mA on the SOA IDAC. The SOA IDAC also comes with a 0 mA to −60 mA current sink capability.

A precision 2.5 V on-chip reference source is also provided. This reference is used by the internal ADC, IDACs, and VDAC circuits, ensuring low drift performance for all of these peripherals. Also provided are two buffered reference outputs capable of sourcing up to 1.2 mA. These can be used externally to the chip.

The ADuCM310 integrates an 80 MHz ARM Cortex-M3 processor, which is a 32-bit RISC machine, offering up to 100 DMIPS peak performance. The ARM Cortex-M3 processor also has a flexible 14-channel DMA controller supporting communication peripherals SPI, UART, and I^{2}C. There are 256 kB of nonvolatile Flash/EE memory and 32 kB of SRAM integrated on-chip.

A 16 MHz on-chip oscillator generates the 80 MHz system clock. This clock can be internally divided for the processor to operate at a lower frequency, thus saving power. A low power, internal, 32 kHz oscillator is available and can clock the timers. There are three general-purpose timers, a wake-up timer, and a system watchdog timer.

A range of communication peripherals can be configured as required in a specific application. These peripherals include UART, two I^{2}Cs, two SPI serial input/output communication controllers, GPIO ports, and PWM.

On-chip factory firmware supports in-circuit serial download via the UART, while nonintrusive emulation and program download is also supported via the serial wire interface. These features are incorporated into a low cost development system supporting this precision analog microcontroller family.

The ADuCM310 operates from 2.9 V to 3.6 V and is specified over a temperature range of −10°C to +85°C.

MAIN FEATURES OF THE ADuCM310

ADC

- Multichannel, 14-bit, 800 kSPS SAR ADC
- 14-bit no missing codes
- Low drift, on-chip voltage reference

DACs

- Eight voltage output DACs
  - VDACs are 12-bit monotonic
- Six current output DACs
  - One of these can source or sink current
  - Current DACs when used as current sources are 14-bit monotonic
- Low drift, on-chip 2.5 V voltage reference source
  - Two buffered reference outputs

Communication

- UART
  - Industry standard, 16450 UART peripheral
  - Support for DMA
- Two I^{2}Cs
  - 2-byte transmit and receive FIFOs for the master and slave
  - Support for DMA
  - Automatic clock stretching option
- Two SPIs
  - Master or slave mode with separate 4-byte Rx and Tx FIFOs
  - Rx and Tx DMA channels
- 8-channel PWM
- Multiple GPIO pins

Processing

- ARM Cortex-M3 processor operating from an internal 80 MHz system clock
- 256 kB Flash/EE memory, 32 kB SRAM
- In-circuit download and debug via serial wire
- On-chip UART download capability

On-Chip Peripherals

- Three general-purpose timers
- Wake-up timer
- Watchdog timer
- 32-element programmable logic array (PLA)

Packages and Temperature Range

- 6 mm × 6 mm, 112-ball BGA package: −10°C to +85°C

Tools

- Low cost development system
- Third-party compiler and emulator tool support

Applications

- Tunable laser modules

MEMORY ORGANIZATION

The ADuCM310 memory organization is described in this section.

Features

- Cortex-M3 memory system features
    - Predefined memory map
    - Support for bit-band operation for atomic operations
    - Unaligned data access
- ADuCM310 on-chip peripherals are accessed via memory mapped registers, situated in the bit band region.
- User memory sizes options:
    - 32 kB SRAM
    - 256 kB Flash/EE memory

There is an on-chip kernel for manufacturer data and in-circuit download.

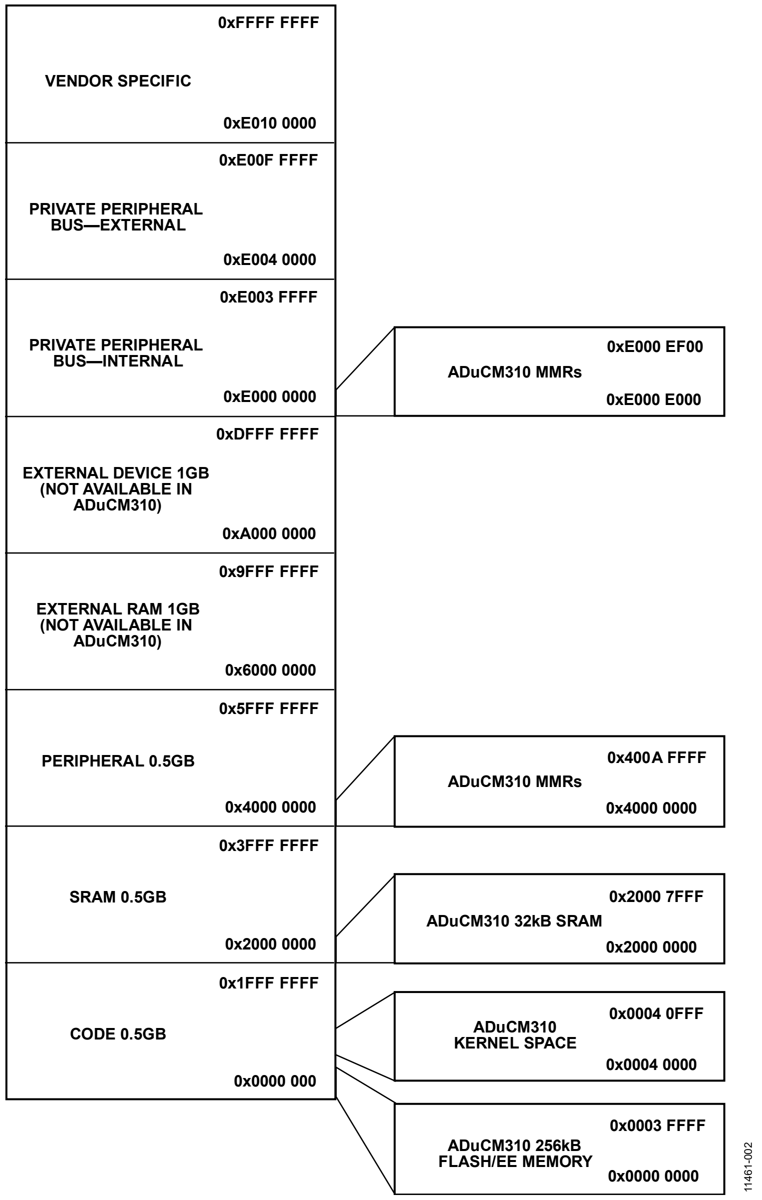


Figure 2. Cortex-M3 Memory Map Diagram

CLOCKING ARCHITECTURE

CLOCKING ARCHITECTURE FEATURES

The ADuCM310 integrates two on-chip oscillators and circuitry for an external crystal and external clock source:

- LFOSC is a 32 kHz, low power internal oscillator that is used in low power modes.
- HFOSC is a 16 MHz internal oscillator that is used in active mode. This is the default input to the PLL.
- HFXTAL is a 16 MHz external crystal oscillator.
- External clock input (ECLKIN) is available via the GPIO pin.

CLOCKING ARCHITECTURE BLOCK DIAGRAM

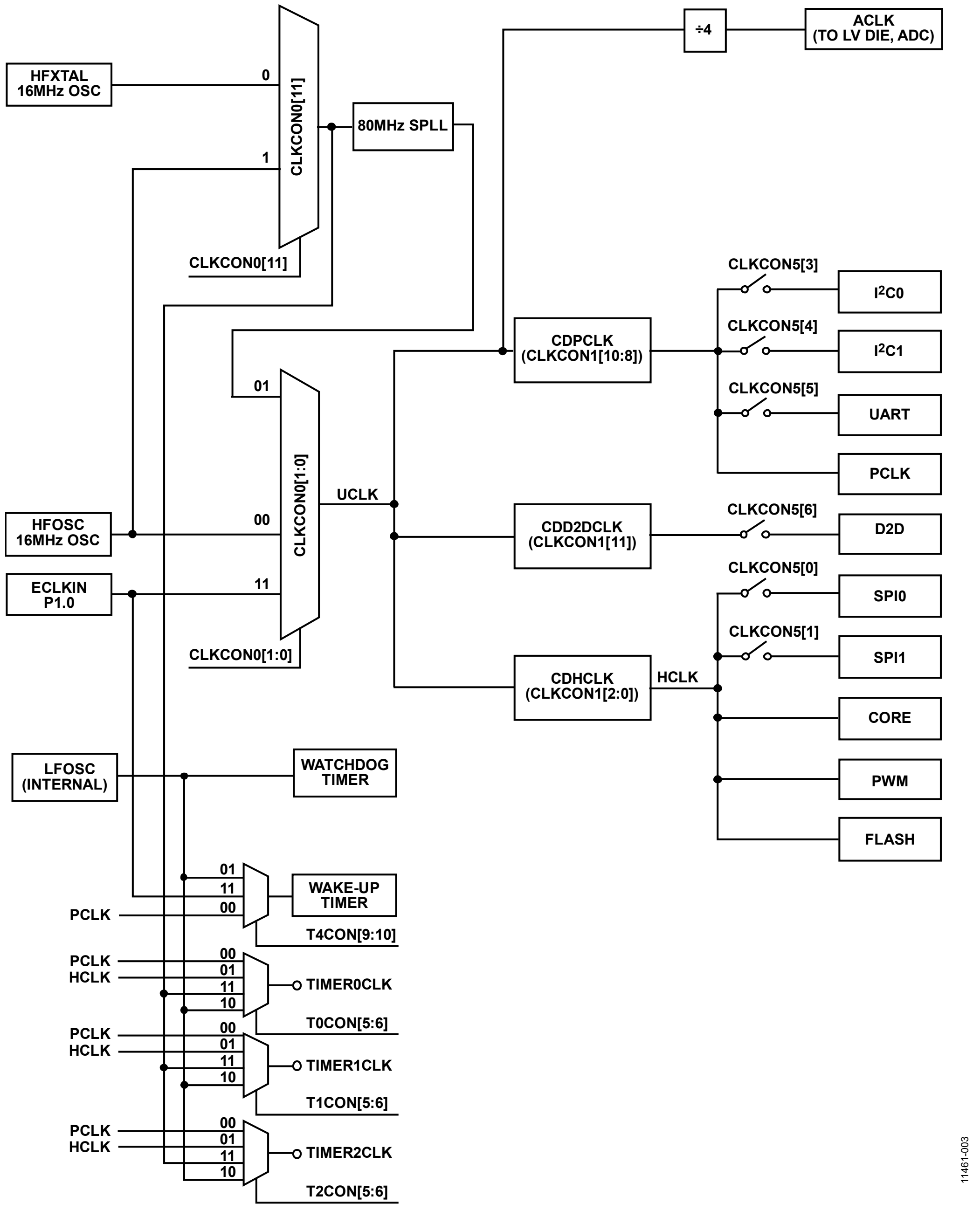


Figure 3. Clocking Architecture Block Diagram

CLOCKING ARCHITECTURE OVERVIEW

The system clock, UCLK can be selected from an 80 MHz PLL output (default). An external clock on P1.0 can also be used for test purposes.

Internally, the system clock is divided into separate clocks:

- UCLK system clock
- System bus clock (HCLK) for the flash, SRAM, and DMA
- ACLK for the analog section of the chip; this is based on the 20 MHz peripheral bus clock (PCLK) output and goes to the low voltage analog die
- PCLK for most peripherals

All ADC performance details are based on a 20 MHz ACLK (CLKCON1[10:8] = 010b). Performance at other clock speeds is not guaranteed; therefore, do not change CLKCON1[10:8] when the ADC is being used.

CLOCKING ARCHITECTURE OPERATION

At power-up, the processor executes at 80 MHz, sourced from the 80 MHz PLL output. The clock source for the 80 MHz PLL is the internal 16 MHz oscillator by default. User code can select the clock source for the system clock and can divide the clock by a factor of 1 to 128, where the clock divider bits are controlled by CLKCON1[2:0]. This allows slower code execution and reduced power consumption.

Note that P1.0 must be configured first as a clock input before the clock source is switched in the clock control register.

If the clock source for the 80 MHz SPLL is required to change from the internal 16 MHz oscillator to the external HFXTAL, observe the following procedure:

1. Check that HFXTAL is stable by reading CLKSTAT0[14:12].
2. Change the system clock to the internal 16 MHz oscillator using CLKCON0[1:0].
3. Switch the input to the SPLL using CLKCON0[11].
4. Wait until the SPLL has locked by monitoring CLKSTAT0[2:0].
5. Change the system clock back to the SPLL clock.

REGISTER SUMMARY: CLOCK ARCHITECTURE

Table 4. Clocking Register Summary

Address	Name	Description	Reset	Access
0x40028000	CLKCON0	Miscellaneous clock settings register	0x0041	RW
0x40028004	CLKCON1	Clock dividers register	0x0200	RW
0x40028014	CLKCON5	User clock gating control register	0x0040	RW
0x40028018	CLKSTAT0	Clocking status register	0x0003	RW

REGISTER DETAILS: CLOCK ARCHITECTURE

Miscellaneous Clock Settings Register

Address: 0x40028000, Reset: 0x0041, Name: CLKCON0

Table 5. Bit Descriptions for CLKCON0

Bits	Bit Name	Description	Reset	Access
15	HFXTALIE	High frequency crystal interrupt enable. 0: an interrupt to the core is not generated on a HFXTAL OK or HFXTAL not OK. 1: an interrupt to the core is generated on a HFXTAL OK or HFXTAL not OK.	0x0	RW
14	UPLLIE	UPLL interrupt enable. 0: UPLL interrupt is not generated. 1: UPLL interrupt is generated.	0x0	RW
13	SPLLIE	SPLL interrupt enable. 0: system PLL interrupt is not generated. 1: system PLL interrupt is generated.	0x0	RW
12	RESERVED	Reserved.	0x0	R
11	PLLMUX	PLL source selection. 0: internal oscillator is selected (HFOSC). 1: external oscillator is selected (HFXTAL).	0x0	RW

Bits	Bit Name	Description	Reset	Access
[10:8]	RESERVED	Reserved.	0x0	RW
[7:4]	CLKOUT	GPIO clock out selection. 0000: UCLK. 0001: LFOSC (32 kHz). 0010: HFOSC (16 MHz). 0100: core clock (default). 0101: PCLK. 1011: General-Purpose Timer 0 clock. 1100: wake-up timer clock. 1110: HFXTAL. All other combinations are reserved.	0x4	RW
[3:2]	RESERVED	Reserved.	0x0	R
[1:0]	CLKMUX	Clock selection. 00: high frequency internal oscillator (HFOSC). 01: SPLL is selected (80 MHz). 10: UPLL is selected (60 MHz). 11: external GPIO port is selected (ECLKIN).	0x1	RW

Clock Dividers Register

Address: 0x40028004, Reset: 0x0200, Name: CLKCON1

Table 6. Bit Descriptions for CLKCON1

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved	0x0	R
[10:8]	CDPCLK	PCLK divide bits. 000: Reserved. 001: Reserved. 010: DIV4. Divide by 4 (PCLK is a quarter the frequency of root clock, 20 MHz). All ADC specifications are based on this setting. Using any other setting may affect ADC performance. 011: DIV8. Divide by 8. 100: DIV16. Divide by 16. 101: DIV32. Divide by 32. 110: DIV64. Divide by 64. 111: DIV128. Divide by 128.	0x2	RW
[7:3]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
[2:0]	CDHCLK	HCLK divide bits. 000: DIV1. Divide by 1 (HCLK is equal to root clock). 001: DIV2. Divide by 2 (HCLK is half the frequency of root clock). 010: DIV4. Divide by 4 (HCLK is quarter the frequency of root clock). 011: DIV8. Divide by 8. 100: DIV16. Divide by 16. 101: DIV32. Divide by 32. 110: DIV64. Divide by 64. 111: DIV128. Divide by 128.	0x0	RW

User Clock Gating Control Register

Address: 0x40028014, Reset: 0x0040, Name: CLKCON5

The user clock gating control register (CLKCON5) controls the gates of the peripheral UCLKs.

Table 7. Bit Descriptions for CLKCON5

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved. Always returns 0 when read.	0x0	R
6	RESERVED	Reserved. Always set to 1.	0x1	RW
5	UCLKUARTOFF	UART clock user control. This bit disables the UCLK UART clock. It controls the gate on UCLK UART in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the UCLK_UART is always off and this bit has no effect. 0: die-to-die clock on permanently. 1: die-to-die clock enabled only during die-to-die transfers (recommended).	0x0	RW
4	UCLKI2C1OFF	I^{2}C1 clock user control. This bit disables the UCLK I^{2}C1 clock. It controls the gate on UCLK I^{2}C1 in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the I^{2}C1 UCLK is always off and this bit has no effect. 0: clock on. 1: clock off.	0x0	RW
3	UCLKI2C0OFF	I^{2}C0 clock user control. This bit disables the UCLK I^{2}C0 clock. It controls the gate on UCLK I^{2}C0 in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the UCLK I^{2}C0 is always off and this bit has no effect. 0: clock on. 1: clock off.	0x0	RW
2	RESERVED	Reserved.	0x0	R
1	UCLKSPI1OFF	SPI1 clock user control. This bit disables the UCLK SPI1 clock. It controls the gate on UCLK SPI1 in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the UCLK SPI1 is always off and this bit has no effect. 0: clock on. 1: clock off.	0x0	RW
0	UCLKSPI0OFF	SPI0 clock user control. This bit disables the UCLK SPI0 clock. It controls the gate on UCLK SPI0 in Power Mode 0 and Power Mode 1. In Power Mode 2 and Power Mode 3, the UCLK SPI0 is always off and this bit has no effect. 0: clock on. 1: clock off.	0x0	RW

Clocking Status Register

Address: 0x40028018, Reset: 0x0003, Name: CLKSTAT0

The clocking status register monitors PLL and oscillator status.

Table 8. Bit Descriptions for CLKSTAT0

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved. Always returns 0 when read.	0x0	R
14	HFXTALNOK	HF crystal not stable. This bit is sticky. It interrupts the core when interrupts are enabled. Write a 1 to this location to clear it. 0: HF crystal stable signal has not been deasserted. 1: HF crystal stable signal has been deasserted.	0x0	RW
13	HFXTALOK	HF crystal stable. This bit is sticky. It interrupts the core when interrupts are enabled. Write a 1 to this location to clear it. 0: HF crystal stable signal has not been asserted. 1: HF crystal stable signal has been asserted.	0x0	RW
12	HFXTALSTATUS	HF crystal status. 0: HF crystal is not stable or not enabled. 1: HF crystal is stable.	0x0	R
[11:3]	RESERVED	Reserved.	0x0	R
2	SPLLUNLOCK	System PLL unlock. This bit is sticky. SPLLUNLOCK is set when the PLL loses its lock. SPLLUNLOCK is used as the interrupt source to signal the core that a lock was lost. Writing a 1 to this bit clears it. SPLLUNLOCK does not set again unless the system PLL gains a lock and subsequently loses it again. 0: no loss of PLL lock was detected. 1: PLL loss of lock was detected.	0x0	RW
1	SPLLLOCK	System PLL lock. This bit is sticky. SPLLLOCK is set when the PLL locks. SPLLLOCK is used as the interrupt source to signal the core that a lock was detected. Writing a 1 to this bit clears it. SPLLLOCK does not set again unless the system PLL loses lock and subsequently locks again. 0: no PLL lock event was detected. 1: PLL lock event was detected.	0x1	RW
0	SPLLSTATUS	System PLL status. Indicates the current status of the PLL. Initially, the system PLL is unlocked. After a stabilization period, the PLL locks and is ready for use as the system clock source. This is a read only bit. A write has no effect. 0: the PLL is not locked or not properly configured. The PLL is not ready for use as the system clock source. 1: the PLL is locked and is ready for use as the system clock source.	0x1	R

POWER MANAGEMENT UNIT

POWER MANAGEMENT UNIT FEATURES

The power management unit (PMU) controls the different power modes of the ADuCM310.

The following four power modes are available:

- Active
- CORE_SLEEP
- SYS_SLEEP
- Hibernate

POWER MANAGEMENT UNIT OVERVIEW

The Cortex-M3 sleep modes are linked to the PMU modes and are described in this section. The PMU is in the always-on section. Each mode provides a power reduction benefit with a corresponding reduction in functionality.

POWER MANAGEMENT UNIT OPERATION

The debug tools can prevent the Cortex-M3 from fully entering its power saving modes by setting bits in the debug logic. Only a power-on reset resets the debug logic. Therefore, the device must be power cycled after using serial wire debug with application code containing the WFI instruction.

Power Mode: Active Mode, Mode 0

The system is fully active. Memories and all user enabled peripherals are clocked, and the Cortex-M3 processor is executing instructions. Note that the Cortex-M3 processor manages the internal clocks and can be in a partial clock gated state. This clock gating affects only the internal Cortex-M3 processing core. Automatic clock gating is used on all blocks. User code can use a WFI command to put the Cortex-M3 processor into sleep mode; it is independent of the power mode settings of the PMU.

When the ADuCM310 wakes up from any of the low power modes, the device returns to Mode 0.

Power Mode: CORE_SLEEP Mode, Mode 1

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

Power Mode: SYS_SLEEP Mode, Mode 2

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

Power Mode: Hibernate Mode, Mode 3

In hibernate mode, the system disables power to all combinational logic and places sequential logic in retain mode. Because FCLK is stopped, the number of sources capable of waking up the system is restricted. The sources listed in Table 64 are the only sources able to wake up the system.

Enter Power Mode 1 to Power Mode 3 when the processor is not in an interrupt handler. If Power Mode 1 to Power Mode 3 is entered when the processor is in an interrupt handler, the power-down mode can be exited only by a reset or a higher priority interrupt source.

CODE EXAMPLES

Code Example to Enter Power Saving Modes

```
SCB->SCR = 0x04;                  // sleepdeep mode
pADI_PWRCTL->PWRKEY = 0x4859;     // key1
pADI_PWRCTL->PWRKEY = 0xF27B;     // key2
pADI_PWRCTL->PWRMOD = 0x3;       // Hibernate
__DSB();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
__WFI();
__nop();
__nop();
__nop();
__nop();
__nop();
__nop();
```

Code Example to Achieve Further Power Savings

```
pADI_ADC->ADCCON = 0;                   // Power off the ADC
pADI_IDAC0->IDACCON = 0x1;              // Turn off IDAC0
pADI_IDAC1->IDACCON = 0x1;              // Turn off IDAC1
pADI_IDAC2->IDACCON = 0x1;              // Turn off IDAC2
pADI_IDAC3->IDACCON = 0x1;              // Turn off IDAC3
pADI_VDAC0->DACCON = 0x100;             // Turn off VDAC0
pADI_VDAC1->DACCON = 0x100;             // Turn off VDAC1
pADI_VDAC2->DACCON = 0x100;             // Turn off VDAC2
pADI_VDAC3->DACCON = 0x100;             // Turn off VDAC3
pADI_VDAC4->DACCON = 0x100;             // Turn off VDAC4
pADI_VDAC5->DACCON = 0x100;             // Turn off VDAC5
pADI_VDAC6->DACCON = 0x100;             // Turn off VDAC6
pADI_VDAC7->DACCON = 0x100;             // Turn off VDAC7
pADI_CLKCTL->CLKCON0 &= 0xFFFC;         // Switch to 16MHz clock
pADI_CLKCTL->CLKCON1 = 0x505;           // Slow down system clocks
pADI_CLKCTL->CLKCON5 = 0x7B;            // Turn off clocks to peripherals
```

REGISTER SUMMARY: POWER MANAGEMENT UNIT

Table 9. Power Management Register Summary

Address	Name	Description	Reset	Access
0x40002400	PWRMOD	Power modes	0x0000	RW
0x40002404	PWRKEY	Key protection for PWRMOD	0x0000	RW

REGISTER DETAILS: POWER MANAGEMENT UNIT

Power Modes Register

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

Table 10. Bit Descriptions for PWRMOD

Bits	Bit Name	Description	Reset	Access
[14:2]	RESERVED	Reserved. These bits must be written 0 by user code.	0x0	R
[1:0]	PWRMOD	Power modes control bits. When read, these bits contain the last power mode value entered by user code. Note that, to place the Cortex in SLEEPDEEP mode for hibernate, the Cortex-M3 system control register (Register 0xE000ED10) must be configured to 0x4 or 0x06. 00: active mode. 01: CORE_SLEEP mode. 10: SYS_SLEEP mode. 11: hibernate mode.	0x0	RW

Key Protection for PWRMOD Register

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

Table 11. Bit Descriptions for PWRKEY

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

ARM CORTEX-M3 PROCESSOR

ARM CORTEX-M3 PROCESSOR FEATURES

High Performance

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

Low Power

- Low standby current.
- Core implemented using advanced clock gating so that only the actively used logic consumes dynamic power.
- Power-saving mode support (sleep and deep sleep modes). The design has separate clocks to allow unused parts of the processor to be stopped.

Advanced Interrupt Handling

- The nested vectored interrupt controller (NVIC) supports up to 240 interrupts, of which the ADuCM310 supports 50. The vectored interrupt feature greatly reduces interrupt latency because there is no need for software to determine which interrupt handler to serve. In addition, there is no need to have software to set up nested interrupt support.
- The ARM Cortex-M3 processor automatically pushes registers onto the stack at the entry interrupt and retrieves them at the exit interrupt. This reduces interrupt handling latency and allows interrupt handlers to be normal C functions.
- Dynamic priority control for each interrupt.
- Latency reduction using late arrival interrupt acceptance and tail-chain interrupt entry.
- Immediate execution of a nonmaskable interrupt request for safety-critical applications.

System Features

- Support for bit-band operation and unaligned data access.
- Advanced fault handling features include various exception types and fault status registers.

Debug Support

- Serial wire debug interfaces (SW-DP).
- Flash patch and breakpoint (FPB) unit for implementing breakpoints. Limited to two hardware breakpoints.
- Data watchpoint and trigger (DWT) unit for implementing watchpoints trigger resources and system profiling. Limited to one hardware watchpoint. The DWT does not support data matching for watchpoint generation because it only has one comparator.

ARM CORTEX-M3 PROCESSOR OVERVIEW

The ADuCM310 contains an embedded ARM Cortex-M3 processor, Revision r2p1. The ARM Cortex-M3 processor 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 outstanding computational performance and exceptional system response to interrupts.

ARM CORTEX-M3 PROCESSOR OPERATION

Several ARM Cortex-M3 processor components are flexible in their implementation. This section details the actual implementation of these components in the ADuCM310.

Serial Wire Debug (SW/JTAG-DP)

The ADuCM310 only supports the serial wire interface via the SWCLK and SWDIO pins. It does not support the 5-wire JTAG interface.

ROM Table

The ADuCM310 implements the default ROM table.

Nested Vectored Interrupt Controller Interrupts (NVIC)

The ARM Cortex-M3 processor includes an NVIC, which offers several features:

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

In addition, the NVIC has a nonmaskable interrupt (NMI) input.

The NVIC is implemented on the ADuCM310, and more details are available in the System Exceptions and Peripheral Interrupts section.

Wake-Up Interrupt Controller (WIC)

The ADuCM310 has a modified WIC, which provides the lowest possible power-down current. More details are available in the Power Management Unit section. It is not recommended to enter a power saving mode while servicing an interrupt. However, if the device does enter a power saving mode while servicing an interrupt, it can be woken up only by a higher priority interrupt source.

μDMA

The ADuCM310 implements the ARM μDMA. More details are available in the DMA Controller section.

ARM CORTEX-M3 PROCESSOR RELATED DOCUMENTS

- Cortex-M3 Revision r2p1 Technical Reference Manual (DDI0337)
- ARM Processor Cortex-M3 (AT420) and Cortex-M3 with ETM (AT425): Errata Notice
- ARMv7-M Architecture Reference Manual (DDI0403)
- ARMv7-M Architecture Reference Manual Errata Markup
- ARM Debug Interface v5 Architecture Specification (IHI 0031)
- PrimeCell® μDMA Controller (PL230) Technical Reference Manual Revision r0p0 (DDI0417)

ADC CIRCUIT

ADC CIRCUIT FEATURES

The ADuCM310 incorporates a fast, multichannel, 16-bit ADC. The ADC is specified to be 14-bit no missing codes.

The flexible input multiplexer supports 10 external inputs and 14 internal channels. The internal channels include the following:

- Temperature sensor channel
- Internal 2.505 V reference
- External reference
- BUF_BREF2.5A and BUF_BREF2.5B
- Six IDAC channels. These are the voltage at each of the IDAC output pins
- PVDD_IDAC2 supply voltage
- $IOV_{DD}/2$ supply voltage.
- $AV_{DD}/2$ supply voltage

The input buffer can be selected for any channel to allow very low input current/input leakage specifications on these input channels. It is recommended to use the input buffer for the $AV_{DD}/2$, $IOV_{DD}/2$, and temperature sensor channels.

The ADC features a high precision, low drift internal 2.505 V reference source.

An external reference can also be connected to the ADC_CAPP and ADC_CAPN pins.

The programmable ADC update rate is from 19.55 kSPS to 800 kSPS.

There is an internal digital comparator for the AIN4 channel. An interrupt can be generated if the digital comparator detects an ADC result above/below a user defined threshold.

Each channel has its own distinct data register for its conversion result. For example, when AIN0 is selected, the result appears in ADCDAT0; if AIN7 is selected, the result appears in ADCDAT7. For a differential measurement, the result always appears in the data register of the positive channel.

ADC CIRCUIT BLOCK DIAGRAM

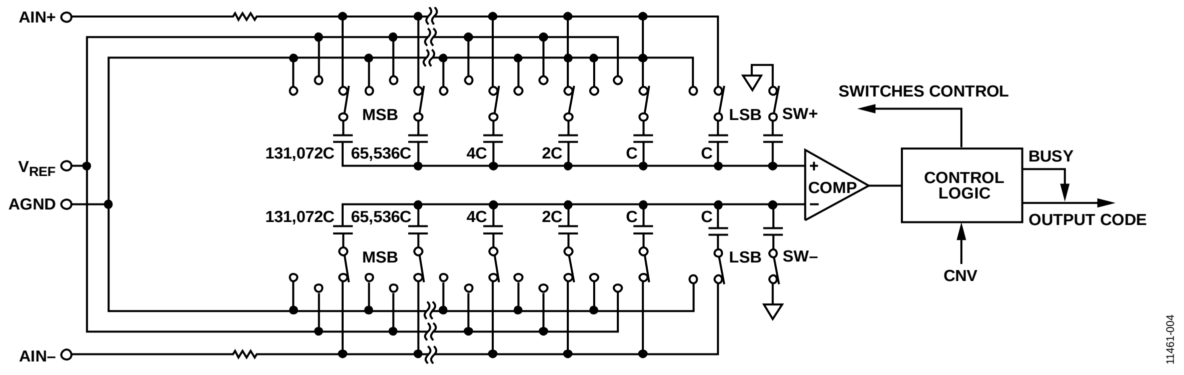


Figure 4. ADC Circuit Block Diagram

ADC CIRCUIT OVERVIEW

The ADuCM310 incorporates a fast, multichannel, 16-bit ADC. The ADC is specified to be 14-bit no missing codes. The ADC can operate from a 2.9 V to 3.6 V supply and is capable of providing a throughput of up to 800 kSPS. This ADC block provides the user with a multichannel multiplexer, input buffer for high impedance input channels, on-chip reference, and SAR ADC.

The SAR ADC circuit is implemented on the low voltage analog die. The ARM Cortex-M3 processor interfaces to the ADC via an internal parallel die-to-die interface.

Depending on the input signal configuration, the ADC can operate in one of the following two modes:

- Differential mode, to measure the difference between two signals.
- Single-ended mode, to measure any signal relative to AGND.

The converter accepts an analog input range of 0 to V_{REF} when operating in single-ended mode. In fully differential mode, the measurement range is $\pm V_{REF}$ while each signal must be between the range of AGND to AV_{DD}.

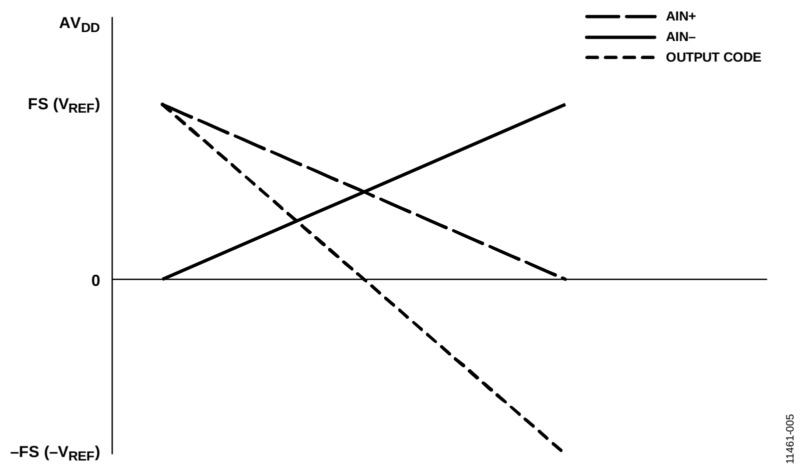


Figure 5. Examples of Balanced Signals for Differential Mode

A high precision, low drift, factory-calibrated 2.505 V reference is provided on-chip. An external reference can also be connected to the ADC_CAPP and ADC_CAPN pins.

Single or continuous conversion modes can be initiated in software. An external pin (alternate function of P2.4) can also generate a repetitive trigger for ADC conversions.

ADC CIRCUIT OPERATION

The SAR ADC is based on a charge redistribution DAC. The capacitive DAC consists of two identical arrays of 18 binary weighted capacitors that are connected to the two inputs of the comparator.

The ADC converts the voltage applied to the positive analog inputs (AIN+) and negative analog inputs (AIN−) in the following three steps:

1. Precharge Phase: during this phase, the precharge buffers connect the inputs to the capacitor arrays. This charges the capacitors quickly with minimal loading of the external input source.
2. Acquisition Phase: during the acquisition phase, the capacitor arrays are connected directly to the inputs to fully charge the capacitor arrays and eliminate any precharge buffer errors. The timing for the acquisition phase is set by ADCCNVC[25:16]; set this value to 1 µs for ADC update rates ≤500 kSPS (for example, ADCCNVC = 0x140028 for 500 kSPS). If the input buffer is not used when measuring $AV_{DD}/2$, $IOV_{DD}/2$, or temperature sensor channels, set this value to 1.5 µs.
3. Conversion Phase: at the end of the acquisition phase, the internal CNV signal goes high and initiates the conversion phase. The conversion begins with the SW+ and SW− switches being opened. This disconnects the two capacitor arrays from the analog inputs and connects the analog inputs to AGND ($-V_{REF}$). The conversion is completed by normal successive approximation.

The ADC block operates from an internally generated 20 MHz clock.

The ADC conversion rate is set by ADCCNVC[9:0].

ADC TRANSFER FUNCTION

Single-Ended Mode

In single-ended mode, the input range is 0 to V_{REF}. The output coding is straight binary with 1 LSB = FS/65,536 or V_{REF}/65,536 = 2.5 V/65,536 = 38.14 µV.

The data values in ADCDATx are aligned such that the MSB is in ADCDATx[27] and, therefore, the LSB is in ADCDATx[12].

The ideal code transitions occur midway between successive integer LSB values (that is, 1/2 LSB, 3/2 LSB, 5/2 LSB, …, FS − 3/2 LSBs). The ideal input/output transfer characteristic is shown in Figure 6.

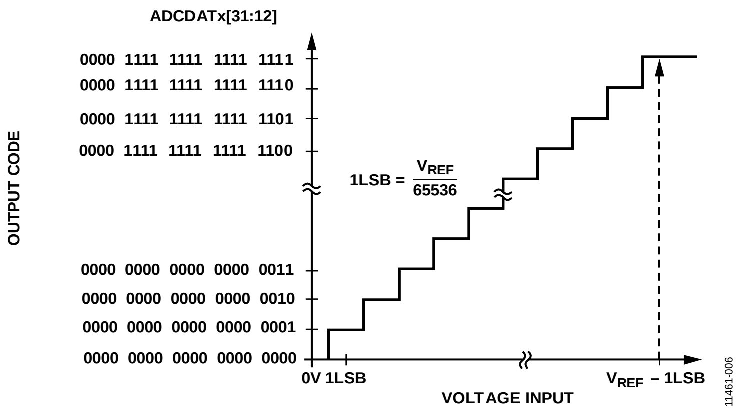


Figure 6. ADC Transfer Function: Single-Ended Mode

Differential Mode

The amplitude of the differential signal is the difference between the signals applied to the positive and negative AINx pins (that is, AIN+ − AIN−). The maximum amplitude of the differential signal is, therefore, $-V_{REF}$ to $+V_{REF}$ p-p ($2 \times V_{REF}$). This is regardless of the common mode (CM). The common mode is the average of the two signals (AIN+ + AIN−)/2, and is, therefore, the voltage that the two inputs are centered on. This results in the span of each input being CM ± $V_{REF}/2$. This voltage must be set up externally, and its range varies with V_{REF}. The voltage at the positive and negative AINx pins must be within the allowed input voltage range.

The output coding is twos complement in fully differential mode, with

1 LSB = 2 V_{REF}/65,536

Or

2 × 2.5 V/65,536 = 76.29 μV

where V_{REF} = 2.5 V.

The ideal input/output transfer characteristic is shown in Figure 7.

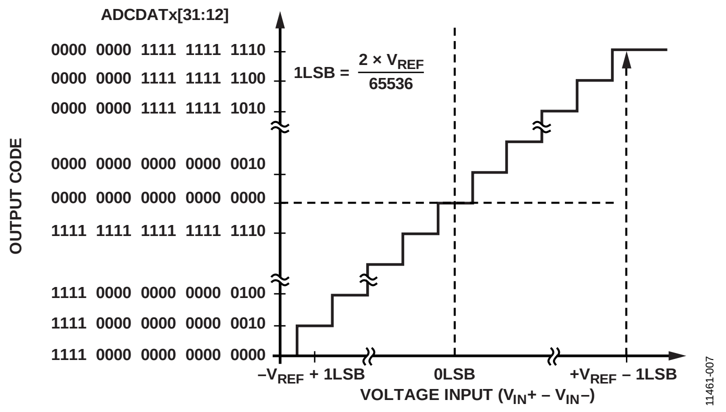


Figure 7. ADC Transfer Function: Differential Mode

ADC TYPICAL SETUP SEQUENCE

After being configured via the ADC control and channel selection registers, the ADC converts the analog input and provides a 16-bit result in the ADC data registers.

The following is an example sequence to set up the ADC and generate a single conversion on AIN0 using a single-ended measurement:

1. Configure the device as follows:

```
ADCCON = 0x280;             // Power up the ADC, enable reference buffer, idle mode.
ADCCHA = 0x1100;            // Select AIN0 as the positive ADC input (AIN+) and
                            VREFN as the negative ADC input (AIN-)
ADCCNVC = 0x1400C8;         // Select 100 kSPS ADC update rate, 500 ns acquisition
                            time.
ADCCON |=0x2;               // Enable single conversion
```

2. Wait for low voltage die interrupt.

```
lADCResult = ADCDAT0;       // read the ADC result
```

Note that if the ADC is set from continuous conversion mode to idle mode after a full ADC sequence is completed, ADCSEQ[31] must be set to 1 before starting another sequence and reconfiguring the ADC back to continuous conversion mode. This ensures that the sequencer restarts with the first selected channel in ADCSEQ.

ADC INPUT BUFFER

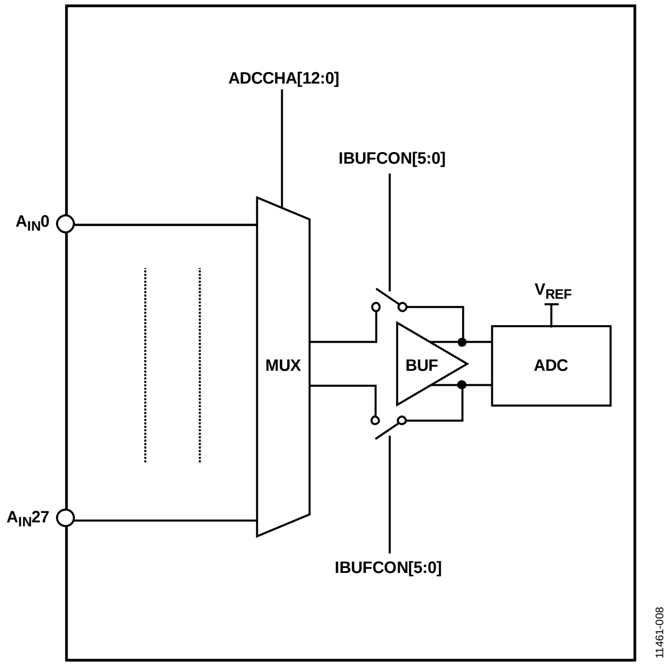


Figure 8. ADC Input Buffer

An optional input buffer can be enabled for any ADC input channel on the ADuCM310.

The IBUFCON control register controls the input buffer switches as follows:

- IBUFCON[1:0] controls the bypass switches on the ADC input buffer. If the input buffer is required on either the positive or negative input, the bypass switch must be turned off.
- IBUFCON[3:2] power up or power down the ADC input buffer.

The input buffer has two options to eliminate the input buffer offset error.

Chop Mode

This mode requires IBUFCON[10] = [1]b. Set IBUFCON[5] = 1 to enable chopping of the positive input buffer.

IBUFCON[4] may be optionally set to enable chopping of the N-side buffer, especially for differential measurements. For single-ended input measurements where the ADC negative input is ground, it is recommended to clear IBUFCON[4] = 0.

Chopping is a common method of offset elimination on input buffers to ADCs.

Chopping requires 2 × ADC conversions per measurement of a buffered ADC input channel.

For example, assume that the input buffer is two input pairs: positive (+) and negative (−) input. V_{IN} applied is 100 mV and the offset on the buffer is 500 µV. Normally, the ADC measures $V_{IN} + V_{OFFSET}$ and reports 100.5 mV when the buffer is enabled. With chopping enabled, the first ADC sample measures $V_{IN} + V_{OFFSET}$ and reports 100.5 mV.

For the second ADC sample of this channel, with chopping enabled, the input pairs are switched and the voltage measured by the ADC is

$$V_{IN} - V_{OFFSET} = 99.5 \text{ mV}$$

Averaging the two results provides the correct result of 100 mV.

To average ADC measurements when the input buffer is enabled, sample an even number of times (that is, 2×, 4×, 6×, and so on).

Auto-Zero Mode

Auto-zero mode does not require averaging so a single ADC measurement gives an accurate result. Signal-to-noise performance of auto-zero mode is slightly degraded compared to chop mode (see Table 1 of the ADuCM310 data sheet for more details).

ADC INTERNAL CHANNELS

Temperature Sensor Settings

The ADuCM310 provides a voltage output from an on-chip band gap reference that is proportional to the absolute temperature of the low voltage die. This voltage output is routed through the front end of the ADC multiplexer (effectively, an additional ADC channel input), facilitating an internal temperature sensor channel that measures die temperature.

The internal temperature sensor is not designed for use as an absolute ambient temperature calculator. The intended use is as an approximate indicator of the temperature of the ADuCM310 low voltage analog die.

An ADC temperature sensor conversion differs from a standard ADC voltage. The ADC performance specifications do not apply to the temperature sensor.

When the temperature sensor channel is selected, the ADC update rate must be <80 kSPS.

- The ADC automatically changes the ADC update rate to 80 kSPS when the temperature sensor, $AV_{DD}/2$, or $IOV_{DD}/2$ input channel is selected. If a different ADC sampling rate is required for other channels after the conversion on any of these three channels is completed, the ADCCNVC register must be updated.
- Note that when the sequencer is enabled and includes any of these three channels, the value in the ADCCNVC does not change and the ADC sampling rate does not change.

The temperature sensor settings are as follows.

To enable the temperature sensor on the ADC, set ADCCHA[12:0] = 0x1116.

To calculate the die temperature, use the following formula:

$$T - T_{REF} = (V_{ADC} - V_{TREF}) \times K$$

where:
T is the temperature result.
T_{REF} is 25°C.
V_{ADC} is the average ADC result from two consecutive conversions.
V_{TREF} is the ADC result in millivolts that corresponds to T_{REF} = 25°C. The user must measure this in their own application because this value varies from device to device. The typical value used for demonstration purposes is 1290 mV.
K is the gain of the ADC in temperature sensor mode. Determine the gain by performing a two-point temperature calibration, because this value varies from device to device. The typical value used for demonstration purposes only is 4.394 mV/°C.

This corresponds to 1/V TC.

Using the default values from the data sheet without any calibration, the equation becomes the following:

$$T - 25°C = (V_{ADC} - 1290) \times 1/K$$

Therefore, assuming V_{ADC} at 25°C = 1290 mV and the slope mV/C as 4.394mV/°C,

$$T = ((V_{ADC} - 1290)/4.394) + 25$$

where V_{ADC} is in millivolts.

See the latest version of the ADuCM310 data sheet for the most up to date figures.

For increased accuracy, perform a two-point calibration at a controlled temperature value.

The values used in this example for V_{TREF} and K are not guaranteed values. The values V_{TREF} and K vary from device to device; therefore, the user must derive the appropriate values by performing a calibration at ambient temperature.

IDAC Channels

The ADuCM310 allows the voltage on the IDAC output pins to be selected as inputs to the ADC for debug purposes.

Take care when selecting the SOA IDAC (IDAC6) as the ADC input; it must never be selected when the SOA shutdown circuit is enabled (sink current source enabled, HVCON[3] = 1).

Applying a negative voltage less than −0.3 V to ADC MUX results in a latch-up condition that is outside the absolute maximum input voltage ratings to the low voltage analog die.

To select the SOA IDAC as an ADC input, HVCON[0] must be set to 1.

AV_{DD}/2 and IOV_{DD}/2 Supply Voltage Channels

These supply voltage channels are measured via internal resistor dividers. Because the resistors used are high impedance and the divided voltage is not buffered, a slower ADC update rate must be used.

The ADC automatically changes the ADC update rate to 80 kSPS when the temperature sensor, AV_{DD}/2, or IOV_{DD}/2 input channels are selected.

If a different ADC sampling rate is required for other channels after the conversion on any of these three channels is completed, the ADCCNVC register must be updated.

Note that when the sequencer is enabled and includes any of these three channels, the value in the ADCCNVC register does not change and the ADC sampling rate does not change. At rates above 80 kSPS, the accuracy is reduced if the input buffer is disabled.

ADC SUPPORT CIRCUITS

ADC Offset and Gain Calibration

For ADC offset calibration, ADCOF[13:0] provides a 14-bit offset calibration. The default value is 0x2000. The calibration resolution is ¼ of the ADCDAT LSB resolution. Therefore, if the 2.505V internal reference is used, the LSB resolution of ADCDAT is ~38.225 µV. In this case, the ADCOF LSB resolution is 9.556 µV.

When performing an offset calibration, apply 0 V to the analog input channel when the input buffer is disabled. If the input buffer is enabled, apply 150 mV.

Similarly, for ADC gain calibration, ADCGN[13:0] provides a 14-bit gain calibration. The default value is 0x2000. The calibration resolution is ¼ of the ADCDAT LSB resolution. Therefore, if the 2.5 V internal reference is used, the LSB resolution of ADCDAT is again~38.225 µV, ADCGN LSB resolution is 9.556 µV. When performing a gain calibration, apply a voltage close to full-scale to the analog input channel. For example, if 2.505 V is the required full-scale value, apply 2.5 V to the input.

ADC Digital Comparator

A digital comparator is provided to allow an interrupt to be triggered if the ADCDAT4 result is above or below a programmable threshold.

Note that only external input channel AIN4 can be used with the digital comparator.

To set up the ADC digital comparator, note the following:

- ADCCMP[17:2] are set a 16-bit ADC threshold value. This value is compared with ADCDAT4[27:12].
- ADCCMP[1] configures the comparator to be triggered when the ADC result is above or below the trigger value.
- To enable the ADC comparator interrupt, set INTSEL[2] = 1 to enable the digital comparator to the Low Voltage Die Interrupt 1 signal.
- Similarly, set INTSEL[10] = 1 to enable the digital comparator interrupt to the Low Voltage Die Interrupt 0 signal.
- The comparator output is asserted when the value in ADCDAT4[27:12] rises above the value in ADCCMP[17:2] if ADCCMP[1] = 1. If ADCDAT4[27:12] remains above ADCCMP[17:2], no further comparator interrupts occur. The interrupt only occurs when the comparator circuit detects a rise above the threshold.
- Similarly, if ADCCMP[1] = 0, the comparator output is asserted when the value in ADCDAT4[27:12] falls below the value in ADCCMP[17:2]. If ADCDAT4[27:12] remains below ADCCMP[17:2], no further comparator interrupts occur. The interrupt only occurs when the comparator circuit detects a fall below the threshold value.

ADC Channel Sequencer

An ADC sequencer is provided to reduce the processor overhead of sampling and reading individual channels. The ADC sequencer allows a user to select the ADC input channels that the ADC samples, and provides a single interrupt source that is asserted when the sequence ends. The sequencer can also be programmed to restart automatically without a delay or a programmable delay between the end and start of sequences.

Some additional details about the sequencer include the following:

- The sequencer reads the ADCSEQ[0:27] register to determine which channels must be included and which must be excluded from the execution sequence.
- ADCSEQ corresponds to ADCCHA[4:0] for the list of ADC input channels. For example, to include AIN9, set ADCSEQ[9].
- To enable the sequencer as the Low Voltage Die Interrupt 1 source, set INTSEL[1] = 1. To enable the sequencer as the Low Voltage Die Interrupt 0 source, set INTSEL[9] = 1.
- To start the sequencer, set ADCSEQ[31:30] = 0x3.
- The ADCSEQC[27:20] bits set the delay between finishing one sequence of channels and starting another sequence.
- Normally, single-ended measurements are assumed by the ADC with AGND as the negative reference. However, for Channel 0, Channel 2, Channel 4, and Channel 6, a differential measurement can be selected by configuring the appropriate bits in ADCSEQC[19:0]. For example, ADCSEQC[4:0] selects the negative input when AIN0 is the positive. For single-ended measurements using the sequencer and AIN0, set ADCSEQC[4:0] to 0x11 for AGND.
- Take care when using the sequencer if the input buffer is enabled. The IBUFCON register controls the input buffer. If the input buffer is enabled, all channels sampled in a sequence are sampled with the input buffer enabled. It is recommended to split sequences into the following:
  - Sample unbuffered channels together in one sequence.
  - Sample buffered channels in a separate sequence. Note that when using the sequencer and the input buffer are enabled with chop mode enabled, ensure that an odd number of channels are enabled in the sequence (for example, 5, 7, or 9). Two consecutive sequences are required with the input buffer enabled with chopping. Average the two results for each enabled channel after the second sequence for the final result.
  - Sample the internal channels ($IOV_{DD}/2$, $AV_{DD}/2$, and internal temperature sensor channels) separately if the ADC update rate selected by ADCCNVC is >100 kSPS. Note that when the sequencer is enabled and includes any of these three channels, the value in ADCCNVC does not change.

Temporarily Halting the ADC Channel Sequencer

It may be required to temporarily halt the ADC channel sequencer before it has fully completed to allow a single or multiple ADC measurements of a channel(s) not included in the sequence.

A use case may be where the internal channels and a number of the external channels must be monitored less frequently than alarm inputs, which must be monitored more regularly. In this case, the sequencer includes all the slower inputs and the sequencer is halted frequently to monitor the alarm input channels. To support this, follow this sequence if ADCSEQS[2] = 1:

1. Check ADCSEQS[2]. If this bit is set to 1, the ADC sequencer is busy and Step 1 to Step 6 must be followed. If ADCSEQS[2] = 0, skip Step 2 to Step 6 and proceed with the single conversions.
2. Set ADCCON[2:0] = [11]b.
3. Set ADCSEQ[29] = 1.
4. Wait for ADCSEQS[1] to clear to 0.
5. After Step 4, clear ADCCON[2:0] = [00]b to disable ADC conversions.
6. Wait for ADCSEQS[0] to clear to 0.
7. After Step 6, the ADC sequencer is properly stalled and the ADC is ready for single conversions.

ADC DMA (Direct Memory Access)

The ADC or the ADC sequencer can be selected as the source channel for the DMA controller. This reduces processor overhead by moving ADC results directly into SRAM with a single interrupt asserted when the required number of ADC conversions has been completely logged to memory.

When using the ADC sequencer with the DMA controller, it is recommended to use DMA automatic request transfer types rather than basic transfer types.

ADC Voltage Reference Selection

The ADuCM310 integrates a low drift, 2.5 V ADC reference source. By default, this internal reference is enabled and selected as the reference source for the ADC. When using the internal 2.5 V voltage reference, ensure the following:

- ADCCON[7] = 1 to power up the internal reference buffer
- AFEREFC[3] = 0 to select the internal reference as the ADC reference source

It is also possible to select an external reference source through the ADC_CAPP pin.

To select an external voltage source as the ADC reference source, ensure the following:

- ADCCON[7] = 0 to power down the internal reference buffer
- AFEREFC[3] = 1 to select the external reference as the ADC reference source

The external reference source must be capable of driving the 4.7 μF capacitor on the ADC_CAPP pin.

If switching from the external to internal reference voltage source, note there is a power-on time specification given in the ADuCM310 data sheet for the ADC reference buffer to fully power up after ADCCON[7] is set to 1.

REGISTER SUMMARY: ADC CIRCUIT

The CPU accesses the ADC circuit over a die-to-die interface (D2D), which increases the execution times of ldr and str instructions. The 32-bit MMRs have 0x40086xxx addresses and require 8 CPU cycles at 80 MHz to execute. The 16-bit MMRs have 0x40082xxx addresses and require 6 CPU cycles at 80 MHz to execute.

Table 12. ADC Circuit Register Summary

Address	Name	Description	Reset	Access
0x40082174	ADCCON	ADC configuration	0x0280	RW
0x40086000	ADCDAT0	ADC0 data and flags	Undefined	R
0x40086004	ADCDAT1	ADC1 data and flags	Undefined	R
0x40086008	ADCDAT2	ADC2 data and flags	Undefined	R
0x4008600C	ADCDAT3	ADC3 data and flags	Undefined	R
0x40086010	ADCDAT4	ADC4 data and flags	Undefined	R
0x40086014	ADCDAT5	ADC5 data and flags	Undefined	R
0x40086018	ADCDAT6	ADC6 data and flags	Undefined	R
0x4008601C	ADCDAT7	ADC7 data and flags	Undefined	R
0x40086020	ADCDAT8	ADC8 data and flags	Undefined	R
0x40086024	ADCDAT9	ADC9 data and flags	Undefined	R
0x4008602C	ADCDAT11	ADC11 data and flags	Undefined	R
0x40086030	ADCDAT12	ADC12 data and flags	Undefined	R
0x40086040	ADCDAT16	ADC16 data and flags	Undefined	R
0x40086044	ADCDAT17	ADC17 data and flags	Undefined	R
0x40086048	ADCDAT18	ADC18 data and flags	Undefined	R
0x4008604C	ADCDAT19	ADC19 data and flags	Undefined	R
0x40086050	ADCDAT20	ADC20 data and flags	Undefined	R
0x40086054	ADCDAT21	ADC21 data and flags	Undefined	R
0x40086058	ADCDAT22	ADC22 data and flags	Undefined	R
0x4008605C	ADCDAT23	ADC23 data and flags	Undefined	R
0x40086060	ADCDAT24	ADC24 data and flags	Undefined	R
0x40086064	ADCDAT25	ADC25 data and flags	Undefined	R
0x40086068	ADCDAT26	ADC26 data and flags	Undefined	R
0x4008606C	ADCDAT27	ADC27 data and flags	Undefined	R
0x40086080	ADCCHA	ADC channel select	0x00111F	RW
0x40086084	ADCSEQS	ADC sequencer stalling status	0x00000001	R
0x40086088	ADCSEQ	ADC sequencer control	0x00000000	RW
0x4008608C	ADCSEQC	ADC sequencer configuration	0x00011111111	RW
0x40086090	ADCGN	User gain adjust	Set during production testing	RW
0x40086094	ADCOF	User offset adjust	0x2000	RW
0x40086098	ADCCMP	Digital comparator configuration	0x00000	RW
0x4008609C	ADCCNVC	ADC conversion configuration	0x000A0014	RW

REGISTER DETAILS: ADC CIRCUIT

ADC Configuration Register

Address: 0x40082174, Reset: 0x0280, Name: ADCCON

Table 13. Bit Descriptions for ADCCON

Bits	Bit Name	Description	Reset	Access
[15:10]	SOFT_RESET	Software reset ADC.	0x00	W
9	PUP	ADC power up. 0: power down. 1: power up.	0x1	RW
8	RESERVED	Reserved.	0x0	R
7	REFB_PUP	ADC reference buffer power up. Must be set to 1 for the ADC to operate normally. 0: power down. 1: power up.	0x1	RW
6	RESTART_ADC	Restart ADC, reset analog part of ADC. Active high. 0: clear to 0 for normal ADC operation. 1: set to 1 to reset the ADC.	0x0	W
5	RESERVED	Reserved.	0x0	R
4	SEQ_DMA	DMA request enable for ADC sequence conversion. 0: clear to 0 to disable ADC sequencer DMA access. 1: set to 1 to enable ADC DMA sequencer access.	0x0	RW
3	CNV_DMA	DMA request enable for ADC nonsequence conversion. 0: clear to 0 to disable ADC DMA access. 1: set to 1 to enable ADC DMA access.	0x0	RW
[2:0]	C_TYPE	ADC conversion type. 00: no conversion/idle mode. 01: DIO pin starts conversion (P2.4). 10: single conversion. 11: continuous conversion (use this mode for the sequencer). 100: PLA conversion.	0x0	RW

ADC Data and Flags Registers

Address: See Table 12, Reset: Undefined, Name: ADCDAT0 to ADCDAT9, ADCDAT11, ADCDAT12 and ADCDAT16 to ADCDAT27

Table 14. Bit Descriptions for ADCDATx

Bits	Bit Name	Description	Reset	Access
[31:4]	ADCx_DAT	ADCx data. The numeric value of the conversion is stored in Bit 12 to Bit 27. Bit 28 to Bit 31 are the extended sign bits. Bit 4 to Bit 11 are always 0. The format is twos complement (signed integer).	0x0	RW
3	RESERVED	Reserved.	0x0	R
2	ADCx_OLD	Flag indicating data has already been read. 0: last data has not been read. 1: last data already read.	0x0	RW
1	ADC_CLIPL	Flag indicating data is clipped at lowest possible value. 0: data valid. 1: low underflow occurred.	0x0	R
0	ADC_CLIPH	Flag indicating data is clipped at highest possible value 0: data valid. 1: high overflow occurred.	0x0	R

ADC Channel Select Register

Address: 0x40086080, Reset: 0x00111F, Name: ADCCHA

ADC channel select register for nonsequence operation.

Table 15. Bit Descriptions for ADCCHA

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
[12:8]	ADCCN	Selects channel for ADC negative input. 0x00: AIN0. 0x01: AIN1. 0x02: AIN2. 0x03: AIN3. 0x04: AIN4. 0x05: AIN5. 0x06: AIN6. 0x07: AIN7. 0x08: AIN8. 0x09: AIN9. 0x0A to 0x0F: reserved. 0x10: VREFP_NADC: connect ADC_REFP to negative input. 0x11: VREFN_NADC: connect ADC_REFN to negative input; use this setting for single-ended measurements. 0x12: AGND. 0x13: PGND. 0x14 to 0x1F: reserved.	0x11	RW
[7:5]	RESERVED	Reserved.	0x0	R
[4:0]	ADCCP	Select ADC channel. 0x0: AIN0. 0x1: AIN1. 0x2: AIN2. 0x3: AIN3. 0x4: AIN4. 0x5: AIN5. 0x6: AIN6. 0x7: AIN7. 0x8: AIN8. 0x9: AIN9. 0xA: reserved. 0xB: BUF_VREF2.5B. 0xC: BUF_VREF2.5A. 0x0D to 0x0F: reserved. 0x10: IDAC4. 0x11: IDAC5. 0x12: IDAC3. 0x13: IDAC1. 0x14: IDAC0. 0x15: IDAC2. 0x16: temperature sensor. 0x17: VREFP_PADC (TBC). 0x18: PVDD_IDAC2. 0x19: $IOV_{DD} \div 2$. 0x1A: $AV_{DD} \div 2$. 0x1B: VREFN_NADC. 0x1C to 0x1F: reserved.	0x1F	RW

ADC Sequencer Control Register

Address: 0x40086088, Reset: 0x00000000, Name: ADCSEQ

Table 16. Bit Descriptions for ADCSEQ

Bits	Bit Name	Description	Reset	Access
31	ST	Sequence restart. Forces sequence to start at first channel when sequence is working. 1: set to 1 to restart the sequencer. Cleared after writing 1.	0x0	W
30	EN	Sequence enable. 1: set to 1 to enable the sequencer.	0x0	W
29	STALL_SEQ	Status bit for ADC sequencer. 0: ADC sequencer is running normally or not enabled. 1: ADC is being stalled. Set as part of the sequencer stall procedure.	0x0	RW
[28:0]	CH	Select channels included in sequence operation. For each channel: 0: channel is skipped. 1: channel is included in the sequence. Each bit corresponds to an ADC channel as defined by ADCCHA[4:0]. For example, a value of 0x33 (00110011) includes AIN0, AIN1, AIN4, and AIN5 in the sequence and excludes all other channels. When using the sequencer with the input buffer enabled and with chop mode enabled, ensure an odd number of channels are enabled in the sequence (for example, 5, 7, or 9). Two consecutive sequences are required with the input buffer enabled with chopping. Average the two results for each enabled channel after the second sequence for the final result.	0x0	RW

ADC Sequencer Stall Status Register

Address: 0x40086084, Reset: 0x00000001, Name: ADCSEQS

Table 17. Bit Descriptions for ADCSEQS

Bits	Bit Name	Description	Reset	Access
[31:3]	RESERVED	Reserved.	0x0000000	R
2	SEQSTA	Sequence status. 1: busy. For a single sequence, the sequencer is converting input channels. For repeated sequences, the sequencer is converting channels or is in the delay period waiting to start another sequence. 0: idle. Sequences have completed.	0x0	R
1	STALLSEQ	Stall sequencer status bit. 0: sequencer is running. 1: sequencer is in the stalling phase.	0x0	R
0	STALLCNV	Stall ADC conversion status bit. 0: ADC single or repeated conversions running. 1: ADC conversions are stalling.	0x1	R

ADC Sequencer Configuration Register

Address: 0x4008608C, Reset: 0x00011111111, Name: ADCSEQC

Table 18. Bit Descriptions for ADCSEQC

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved.	0x00	R
[27:20]	T	Define programmable delay of 0 to 254 between sequences. A delay of 255 causes a halt after one sequence. Set ADCSEQ[30] if another sequence is required. If the ADC input buffer is used in chop mode, ensure to use a value of <255.	0xFF	RW
[19:15]	DIF6	Selects differential mode negative input for AIN6 in the sequence. 0x11: Channel 6 is single-ended.	0x11	RW
[14:10]	DIF4	Selects differential mode negative input for AIN4 in the sequence. 0x11: Channel 4 is single-ended.	0x11	RW

Bits	Bit Name	Description	Reset	Access
[9:5]	DIF2	Selects differential mode negative input for AIN2 in the sequence. 0x11: Channel 2 is single-ended.	0x11	RW
[4:0]	DIF0	Selects differential mode negative input for AIN0 in the sequence. 0x11: Channel 0 is single-ended.	0x11	RW

User Gain Adjust Register

Address: 0x40086090, Reset: Set during production testing, Name: ADCGN

Table 19. Bit Descriptions for ADCGN

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R
[13:0]	GN	User gain register.	Set during production testing	RW

User Offset Adjust Register

Address: 0x40086094, Reset: 0x2000, Name: ADCOF

Table 20. Bit Descriptions for ADCOF

Bits	Bit Name	Description	Reset	Access
[13:0]	OF	User offset register.	0x2000	RW

Digital Comparator Configuration Register

Address: 0x40086098, Reset: 0x00000, Name: ADCCMP

Table 21. Bit Descriptions for ADCCMP

Bits	Bit Name	Description	Reset	Access
[17:2]	THR	Compare threshold. Value to compare to Channel 4 data.	0x0000	RW
1	DIR	Select comparator direction. 0: ADCTH less than Channel 4 data. 1: ADCTH larger than Channel 4 data.	0x0	RW
0	EN	Digital comparator enable. 0: disable. 1: enable.	0x0	RW

ADC Conversion Configuration Register

Address: 0x4008609C, Reset: 0x000A0014, Name: ADCCNVC

When ADCCP is set to 22 (temperature sensor), 25 ($IOV_{DD}/2$), or 26 ($AV_{DD}/2$), the ADCCNVC register automatically changes to 0x7D00FA – 80 kSPS for single conversions. Set ADCCNVC to the required conversion rate after sampling these three channels if a different sample rate is required for other ADC input channels.

When the sequencer is enabled and includes any of these three channels, the value in ADCCNVC does not change and the ADC sampling rate does not change.

Table 22. Bit Descriptions for ADCCNVC

Bits	Bit Name	Description	Reset	Access
[31:26]	Reserved	Write 0 to these bits.	0x0	RW
[25:16]	CNVD	Configure ADC acquisition time. Acquisition time = CNVD/20 MHz. Default acquisition time is 500 ns. For best SNR results, ensure that the acquisition time is set to 1 µs (0x14) for ADC conversion rates ≤500 kSPS. Set to 650 ns (0xD) for conversion rates >500 kSPS	0x0A	RW
[15:10]	Reserved	Do not overwrite.	0x0	RW
[9:0]	CNVC	Configure conversion frequency. Conversion frequency = 20 MHz/CNVC.	0x14	RW

REGISTER SUMMARY: ADDITIONAL REGISTERS

Table 23. Register Summary

Address	Name	Description	Reset	Access
0x40083400	IBUFCON	Input buffer control bit	0x000F	RW
0x40087834	AFEREFC	Reference configuration register	0x00	RW

REGISTER DETAILS: ADDITIONAL REGISTERS

Input Buffer Control Bit Register

Address: 0x40083400, Reset: 0x000F, Name: IBUFCON

Table 24. Bit Descriptions for IBUFCON

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	RW
10	IBUF_AZ	Control of auto-zero mode. 0: auto-zero mode enabled. 1: auto-zero mode disabled.	0x0	RW
[9:8]	IBUF_CHOPPOL	Control bits for manual control of input buffer chop switches. 00: (P, N) input buffers offset polarity is (+, +). 01: (P, N) input buffers offset polarity is (+, −). 10: (P, N) input buffers offset polarity is (−, +). 11: (P, N) input buffers offset polarity is (−, −). Clear these bits to 0 if IBUFCON[10] = 0, or if IBUFCON[5:4] > 0.	0x0	RW
[7:6]	RESERVED	Reserved.	0x0	RW
[5:4]	IBUF_CHOP	Chopping mode enable for the input buffers. 00: chopping disabled on both buffers. 01: chopping enabled on N-side only; P-side disabled. 10: chopping enabled on P-side only; N-side disabled. 11: chopping enabled on both buffers.	0x0	RW
[3:2]	IBUF_PD	Power down P/N input buffer separately. 00: both sides powered on. 01: N-side powered down. 10: P-side powered down. 11: both sides powered down	0x3	RW
[1:0]	IBUF_BYP	Bypass P/N input buffer separately. 00: no bypass. 01: N-side bypassed. 10: P-side bypassed. 11: bypass both.	0x3	RW

Reference Configuration Register

Address: 0x40087834, Reset: 0x00, Name: AFEREFC

Table 25. Bit Descriptions for AFEREFC

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved.	0x0	RW
6	AFE_2MA_PDA	Power down the 2 mA Reference Output Driving Buffer A. 0: power up 2.5 V Reference Output Driving Buffer A. 1: power down 2.5 V Reference Output Driving Buffer A.	0x0	RW
[5:4]	RESERVED	Reserved.	0x0	RW
3	AFE_REF_EXT	Select reference source for buffered reference outputs. 0: select internal 2.5 V reference. 1: select external 2.5 V reference.	0x0	RW
2	AFE_2MA_PDB	Power down the 2 mA Reference Output Driving Buffer B. 0: power up 2.5 V Reference Output Driving Buffer B. 1: power down 2.5 V Reference Output Driving Buffer B.	0x0	RW
1	AFE_2V5R_PD	2.5 V reference buffer power down. 0: power up 2.5 V reference buffer. 1: power down 2.5 V reference buffer.	0x0	RW
0	AFE_BG_PD	Band gap power down. 0: power up 1.2 V band gap. 1: power down 1.2 V band gap.	0x0	RW

IDACs

IDAC FEATURES

The ADuCM310 provides six IDACs: five high current DAC sources and one current source and current sink (IDAC3) channel. The current sources are low noise, low drift current outputs.

- IDAC0: 0 mA to 100 mA full-scale output.
- IDAC1: 0 mA to 100 mA full-scale output.
- IDAC2: 0 mA to 200 mA full scale output.
- IDAC3: 0 mA to 250 mA full-scale current source output (using IDAC3CON and IDAC3DAT). It can also be set as a current sink capable of sinking up to 80 mA (using IDAC6CON and IDAC6DAT).
- IDAC4: 0 mA to 20 mA full-scale output.
- IDAC5: 0 mA to 20 mA full-scale output.

IDAC BLOCK DIAGRAM

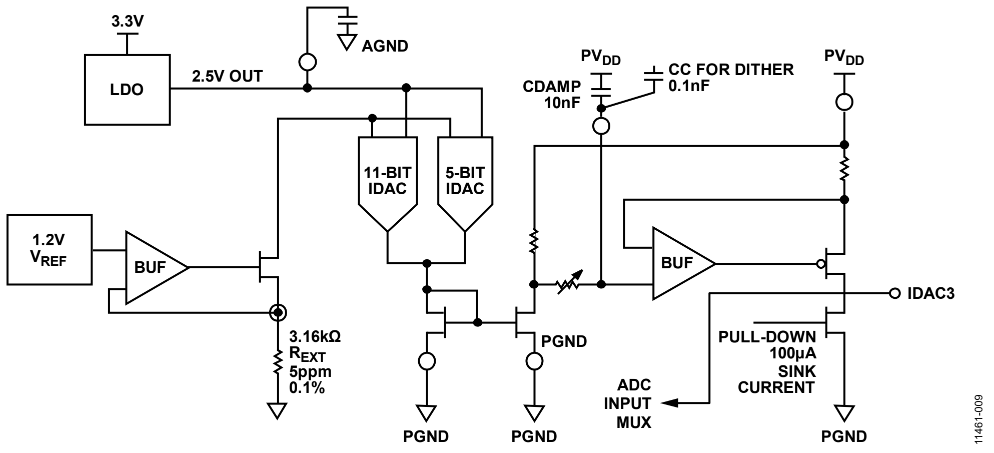


Figure 9. Example IDAC Circuit—IDAC3

IDAC OVERVIEW

Precision Current Generation and Fault Protection

The reference current for the IDACs is generated by a precision internal band gap voltage reference ($V_{BANDGAP}$) and an external precision resistor (R_{REF}, 5 ppm, 0.1%). The reference current is equal to $V_{BANDGAP}$ divided by R_{REF}. The band gap voltage reference is a low drift, high accuracy voltage source that helps to minimize the overall IDAC gain error and gain error drift. The noise of the IDAC outputs is limited by the low pass filter on the output stage; each IDAC requires a 10 nF capacitor between PV_{DD} supply voltage and the CDAMP pin.

Figure 9 shows the typical architecture of the IDAC. The parallel 11-bit and 5-bit IDACs set the output current scale. The output of these IDACs are summed together and fed to a current mirror and then are gained up at the output stage.

Production trimming of the LDO band gap reference aids performance. In addition, gain trimming and scaling of the current mirror and output stages are also included in the ATE test program.

IDAC3 Shutdown

IDAC3 with IDAC6 are designed to control a semiconductor optical amplifier (SOA) stage of a tunable laser. IDAC3 is the main current source DAC. IDAC6 is a selectable current sink that is connected to the same external pins as IDAC3; the pin name in the ADuCM310 data sheet is IDAC3.

The SOA IDAC (IDAC3) is different from the other five IDACs because it can also be configured as a current sink.

The SOA IDAC (IDAC3) has two separate data registers: IDAC3DAT and IDAC6DAT.

- The pull-down stage shown in Figure 9 is controlled by an 11-bit IDAC; the data register for this IDAC is IDAC6DAT, which allows the sink current to be set from 0 mA to 80 mA. When the current sink is enabled, the SOA pin voltage is pulled negative by the high voltage analog die to drain the current out of the SOA diode. To protect the SOA diode, an internal clamp circuit is in place to ensure that the voltage seen by the diode never drops below −3.5 V. The absolute minimum pull-down voltage is −3.7 V.
- IDAC3DAT is the normal current source data register, which sets the IDAC output when used as a current source in the range of 0 mA to 250 mA.
- HVCON is the user control register for the high voltage die. HVCON[1] must be cleared to 0 to enable the SOA sink current. Setting HVCON[1] to 1 disables the SOA sink current on the high voltage die.
- HVCON[11] controls the SOA shutdown clamping voltage. The SOA clamping voltage is controlled in two ways:
  - When HVCON[11] = 0, the default negative shutdown voltage is clamped at approximately −1.75 V.
  - When HVCON[11] = 1, the full trimmed clamping circuit is enabled, meaning the shutdown voltage is clamped in the range of −3.0 V to −3.5 V.

To avoid an undershoot when shutting down the SOA IDAC and enabling the current sink, observe the following procedure to ensure fast and safe shutdown of the SOA IDAC:

1. Enable SOA current sink via IDAC6CON.
2. Configure negative shutdown current via IDAC6DAT.
3. Disable SOA current source, IDAC3CON = 0x3.

These three steps result in a negative voltage on the SOA pin with a voltage setting of greater than −1.75 V.

Next, turn on the full clamping circuit by setting HVCON[11] = 1, which results in the shutdown voltage dropping to −3.0 V to −3.5 V.

For normal SOA operation, when IDAC3 is used as a current source and when there is no requirement for sinking current from the SOA diode, ensure that IDAC6 is configured as follows:

- IDAC6CON = 0x80
- IDAC6DAT = 0x00

Not configuring the IDAC6 registers as shown previously results in an offset error on IDAC0, IDAC1, IDAC4, and IDAC5.

If any of the IDACs are unused, the recommended setting is to power it on and set the data code to 0:

- IDACxCON = 0x80
- IDACxDAT = 0x00

IDAC0, IDAC1, IDAC2, IDAC4, and IDAC5 also have a small current sink capability to minimize the positive offset current when the data register is set to 0. The IDACxCON[1] bit can enable a pull-down current source to PGND. This pull-down current is typically 100 µA.

IDAC Thermal Shutdown

The ADuCM310 has an internal temperature sensor that monitors the die temperature. This temperature sensor can be monitored as an ADC input channel; the measured voltage is proportional to die temperature. See the Temperature Sensor Settings section for more information.

Internally, the die temperature is compared to a fixed voltage, proportional to approximately 130°C die temperature. If the die temperature exceeds 130°C, there is a risk of damaging the die because the absolute maximum junction temperature rating is 150°C. As the IDACs potentially consume the most power, the user may want to shut off the IDACs to reduce power and therefore reduce the die temperature. Two options are available:

- Enable a thermal interrupt by setting either INTSEL[12] or INTSEL[4]. If the die temperature exceeds the threshold of approximately 130°C, this interrupt is triggered and user code can take the appropriate action. This is the recommended procedure.
- Another option is to enable automatic shutdown of the IDACs by setting the individual thermal shutdown bits for each IDAC via IDACxCON[6]. If this bit is set in the appropriate IDACxCON register, the IDAC output current is reduced to 0 mA to reduce the power consumption of the device and to reduce the die temperature of the device.

Note that the internal temperature sensor accuracy can be up to ±20°C and there is no way of calibrating the thermal shutdown trip point. Therefore, it is recommended not to enable the automatic thermal shutdown feature (IDACxCON[6] = 0) without also enabling one of the thermal interrupt sources (INTSEL[4] = 1 or INTSEL[12] = 1).

IDAC Reference Resistor Error Shutdown

The IREF pin is connected to ground via a 3.16 kΩ resistor to generate a reference current for the IDACs. The value of this resistor directly affects the output current on the IDACs. If this resistor is less than 3.16 kΩ, the output current on all IDACs is higher than expected; the output current scales with the resistor value.

The IDAC hardware has a fault detection circuit that detects if the IREF resistor is <50% of 3.16 kΩ (<1.58 kΩ). If the fault is detected on the reference resistor, an interrupt can be generated.

- If INTSEL[5] = 1, this fault detection circuit can generate an interrupt from the low voltage die and a status register is set to indicate the fault.
- While R_{EXT} is detected as <1.58 kΩ, the IDAC outputs are automatically shut down (0 mA output) and remain off until R_{EXT} rises above the 1.58 kΩ threshold.

IDAC Output Filter

Each IDAC has a filter on the output stage to minimize noise. Each IDAC requires an external 10 nF capacitor between the PV_{DD} supply and its CDAMP pin (see the ADuCM310 data sheet). The on-chip, programmable resistor is controlled by the IDACxCON[5:2] bits.

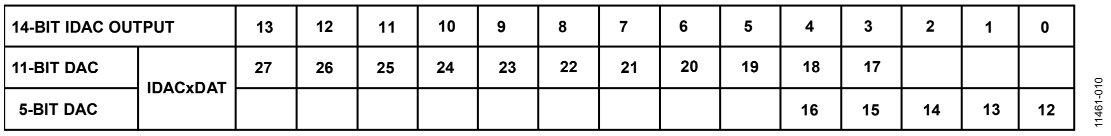
Table 26 IDAC Filter Bandwidth Control Settings

IDACxCON[5:2]	R Value	Cutoff Frequency (f_C)
0000	60 Ω	262 kHz
0101	5.6 kΩ	2.8 kHz
0110	11.2 kΩ	1.4 kHz
0111	22.2 kΩ	715 Hz
1000	44.4 kΩ	357 Hz
1001	104 kΩ	153 Hz
All other options are reserved		

IDAC Data Register

The IDAC output is controlled by an internal 11-bit and 5-bit DAC.

The 11-bit DAC (IDACxDAT[27:17]) controls the MSBs. The 5-bit DAC (IDACxDAT[16:12]) controls the LSBs. Bits[4:3] of the 5-bit DAC overlap the high order 11-bit DAC. The two MSBs of the 5-bit DAC (IDACxDAT[16:15]) overlap the two LSBs of the 11-bit DAC (IDACxDAT[18:17]), as shown in Figure 10.



14-BIT IDAC OUTPUT		13	12	11	10	9	8	7	6	5	4	3	2	1	0
11-BIT DAC	IDACxDAT	27	26	25	24	23	22	21	20	19	18	17			
5-BIT DAC	IDACxDAT										16	15	14	13	12

Figure 10. 14-Bit IDAC Output

The 11-bit DAC and the 5-bit DAC are guaranteed monotonic as individual DACs. However, when combined, there is mismatch between the bit weightings of the two DACs; therefore, when combined, they are not monotonic and not even 12-bit monotonic. Nonlinearity errors occur at certain major code transitions.

To improve the linearity performance when combining the two DACs, the following happens during production testing:

- The ATE production test measures the IDAC output at each of the known major code transitions that log the nonlinearity errors for IDAC0 to IDAC5 on every device.
- The ATE production test logs the correction factor to flash tables beginning at Address 0x40980 to Address 0x40BEF.
- The correction factors are effectively an adjustment of the 5-bit DAC to make a single, linear 14-bit DAC output.
- The tables are organized in three parts for each IDAC:
    - Correction values for IDACxDAT[27:23], or the five MSBs of the 11-bit DAC.
    - Correction values for IDACxDAT[22:21], or Bits[6:5] of the 11-bit DAC.
    - Correction values for IDACxDAT[20:17], or the 4 LSBs of the 11-bit DAC.
    - The following is an example of how to define a structure for this table in source code.

```
typedef struct
    {                            /*IDAC correction structure        */
    unsigned short  pusErr5[6][32];         /*Top 5 bits correction array.        */
    unsigned short  pusErr2[6][4];          /*Next 2 bits correction array.      */
    unsigned short  pusErr4[6][16];         /*Next 4 bits correction array.        */
    } IDACCOR_TypeDef;

#define pIDACCOR    ((IDACCOR_TypeDef *)0x40980)
```

- The correction factor is the value that must be added or subtracted from the ideal value for the IDACxDAT register to generate the required output current.

The ideal value is the IDACxDAT value if the output transfer function had no nonlinearity errors.

For example, setting IDAC2 to 12.527 mA, the ideal 14-bit value to write to the IDAC is 0x2815. (12.527/20 × FS). It is assumed the 14-bit value is comprised of the full 11-bits of the 11-bit IDAC plus the lower 3-bits of the 5-bit IDAC. The 11-bit IDAC is comprised of three sub-DACs. Therefore, when correcting for example IDAC0, three lookup tables are required.

Working with the ideal value first, take the following steps:

1. Identify the ideal 14-bit value first and subtract the worst-case error from this (0x20), assuming that the ideal 14-bit value is >0x20.
2. Index correction factor table for Bits[13:9]. This involves reading the correct entry in the first lookup table to correct 5 MSBs (10:6) of the 11-bit DAC. For example:

   ```
   iErr = pIDACCOR->pusErr5[iChan][(iVal>>9)&0x1f];
   ```

3. Index correction factor for Bits[8:7]. This involves reading the correct entry in the second lookup table for Bits[5:4]of the 11-bit DAC. For example:

   ```
   iErr += pIDACCOR->pusErr2[iChan][(iVal>>7)&3];
   ```

4. Index correction factor for Bits[6:3]. This involves reading the correct entry in the third lookup table for Bits[3:0]of the 11-bit DAC. For example:

   ```
   iErr += pIDACCOR->pusErr4[iChan][(iVal>>3)&0xf];
   ```

5. Compute the final error correction value (see the following example).
6. Add the final error to the IDACxDAT register.

The following example shows how to implement this:

```
int IdacOutC(int iChan, int iIdealVal)
    {
    ADI_IDAC_TypeDef *psIDAC[6] =
{pADI_IDAC0,pADI_IDAC1,pADI_IDAC2,pADI_IDAC3,pADI_IDAC4,pADI_IDAC5};
    unsigned int iError;      //Error as it is processed.
    int iSh = 0x20;
    do
        {
        iSh -= 0x08;
        iError = IdacCor(iChan,iIdealVal,iSh);  //Get correction.
        }
    while(iError>31 && iSh>0);
    if (iIdealVal >=iSh)
      psIDAC[iChan]->IDACDAT = (((iIdealVal-iSh)&0x3FF8)<<14) + (iError<<12);  //Output
corrected value.
    else
       psIDAC[iChan]->IDACDAT = 0 + (iError<<12);  //Output corrected value.
    return  1;
    }
int IdacCor(int iChan, int iIdealVal, int iShift)
    {
    int iVal;               //14 bit value.
    int iErr;      //Error as it is processed.

    iVal = iIdealVal-iShift;    //Reduce 11 bit DAC by fraction of minor DAC.
        //Add together errors from 11 bit DAC.
    iErr = pIDACCOR->pusErr5[iChan][(iVal>>9)&0x1f];
    iErr += pIDACCOR->pusErr2[iChan][(iVal>>7)&3];
    iErr += pIDACCOR->pusErr4[iChan][(iVal>>3)&0xf];
    iErr = ((iVal&0x7)*256 + iErr + iShift*256 - 0x8000*3)/256; //Minor val + error + shift.
    return iErr;
    }
```

IDACs: Common Use Cases

Case 1: Setting the Output Current of IDAC1 to Quarter Scale

Set up IDAC1CON as follows:

- IDAC1CON[7] = 1: enable writes to the IDAC1DAT register.
- IDAC1CON[0] = 0: power up IDAC1.
- IDAC1CON[5:2] as per Table 26 to set up the filter bandwidth as required.
- IDAC1CON[1] = 0: disable the IDAC1 pull-down current source.
- IDAC1DAT[1] = 0: clear the IDAC1 sync bit to allow immediate updating of the IDAC.
- IDAC1CON[6] = 0: disable the overtemperature shutdown feature.

Set up IDAC1DAT to give a quarter-scale current output.

To calculate the correct value to write to IDAC1DAT, take the following steps:

- Correction tables are provided in flash with error correction values for IDAC1.
- As described in the IDAC Data Register section, appropriate reads from each of the lookup tables for IDAC1 are required to adjust for nonlinearity errors in each of the four subsections of IDAC1.
- Adjust the corrected value into the format required to load IDAC1DAT.

Case 2: Switch the SOA IDAC from Sourcing Current to Sinking 40 mA

Set up the IDAC6CON register (SOA current sink control register):

- IDAC6CON[7] = 1: enable writes to the IDAC6DAT register.
- IDAC6CON[0] = 0: power up SOA sink current source.
- HVCON[1] = 0: switch on the SOA sink current on the high voltage die.
- Ensure that HVCON[11] = 0 to avoid a negative voltage undershoot when the shutdown is enabled.

Set up the IDAC3CON register:

- IDAC3CON[1:0] = [11] to disable the SOA current source.
- IDAC3DAT[5] = 0: clear the IDAC3 sync bit to allow immediate updating of the IDAC.

Set up the current sink value:

- Set the 11-bit IDAC to midscale (40 mA) by writing IDAC6DAT[27:17] = 0x800.
- Set the IDAC3 to zero scale by writing IDAC3DAT[27:12] = 0x0000. This sets the current source to 0 mA.
- Set HVCON[11] = 1 to pull the shutdown voltage to the lowest trimmed value.

Case 3: Turn On IDAC2 and Set the Output to 0 mA with the Lowest Possible Offset

Set up the IDAC2CON register:

- IDAC2CON[7] = 1: enable writes to the IDAC2DAT register.
- IDAC2CON[0] = 0: power up IDAC2.
- IDAC2CON[5:2] as per Table 26 to set up the filter bandwidth as required.
- IDAC2CON[1] = 0: enable the IDAC2 pull-down current source.
- IDAC2CON[6] = 0: disable the overtemperature shutdown feature.

Set up the IDAC2DAT register:

- IDAC2DAT[27:12] = 0x0000: set the IDAC to zero scale.

Case 4: Set Up All Six IDACs with their Output Current Updating Simultaneously

Set up the IDACs as per previous cases, except this time set IDACx_SYNC[x] = 1 in each of the IDACxDAT registers.

After writing to the last IDACxDAT and IDACxCON register, set IDAC1DAT[5:0] = [000000]b.

This ensures that all six of the current source IDACs are updated at the same time.

REGISTER SUMMARY: IDAC

Table 27. IDAC Register Summary

Address	Name	Description	Reset	Access
0x40086800	IDAC0DAT	IDAC0 data register	0x00000000	RW
0x40086804	IDAC0CON	IDAC0 control register	0x01	RW
0x40086808	IDAC1DAT	IDAC1 data register	0x00000000	RW
0x4008680C	IDAC1CON	IDAC1 control register	0x01	RW
0x40086810	IDAC2DAT	IDAC2 data register	0x00000000	RW
0x40086814	IDAC2CON	IDAC2 control register	0x01	RW
0x40086818	IDAC3DAT	IDAC3 data register	0x00000000	RW
0x4008681C	IDAC3CON	IDAC3 control register	0x01	RW
0x40086820	IDAC4DAT	IDAC4 data register	0x0000000000	RW
0x40086824	IDAC4CON	IDAC4 control register	0x01	RW
0x40086828	IDAC5DAT	IDAC5 data register	0x00000000	RW
0x4008682C	IDAC5CON	IDAC5 control register	0x01	RW
0x40086830	IDAC6DAT	IDAC6 data register	0x00000000	RW
0x40086834	IDAC6CON	IDAC6 control register	0x01	RW

REGISTER DETAILS: IDAC

IDAC0 Data Register

Address: 0x40086800, Reset: 0x00000000, Name: IDAC0DAT

Table 28. Bit Descriptions for IDAC0DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC0 high data.	0x0	RW
[16:12]	DATL	IDAC0 low data.	0x0	RW
[11:6]	RESERVED	Reserved.	0x0	R
[5:0]	SYNC	IDAC0 sync bits. These six bits are common to the six IDACs. Each bit set to 1 prevents the corresponding channel from updating immediately. The channel updates when the bit changes to 0.	0x00	RW

IDAC0 Control Register

Address: 0x40086804, Reset: 0x01, Name: IDAC0CON

Table 29. Bit Descriptions for IDAC0CON

Bits	Bit Name	Description	Reset	Access
7	CLRB	IDAC0 clear bit. 0: clear IDAC1DAT. 1: enable write.	0x0	RW
6	SHT_EN	IDAC0 shutdown enable. Enables automatic shutdown in case of overtemperature. 0: disable this function. 1: enable this function.	0x0	RW
[5:2]	BW	IDAC0 bandwidth. See the IDAC Output Filter section for more details.	0x0	RW
1	PUL	IDAC0 pull down. 0: disable the pull-down current source. 1: enable the pull-down current source.	0x0	RW
0	PD	IDAC0 power down. 0: powers up IDAC0. 1: powers down IDAC0.	0x1	RW

IDAC1 Data Register

Address: 0x40086808, Reset: 0x00000000, Name: IDAC1DAT

Table 30. Bit Descriptions for IDAC1DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC1 high data.	0x0	RW
[16:12]	DATL	IDAC1 low data.	0x0	RW
[11:6]	RESERVED	Reserved.	0x0	R
[5:0]	SYNC	IDAC1 synchronization bits. These six bits are common to the six IDACs. Each bit set to 1 prevents the corresponding channel from updating immediately. The channel updates when the bit changes to 0.	0x00	RW

IDAC1 Control Register

Address: 0x4008680C, Reset: 0x01, Name: IDAC1CON

Table 31. Bit Descriptions for IDAC1CON

Bits	Bit Name	Description	Reset	Access
7	CLRB	IDAC1 clear bit. 0: clear IDAC1DAT. 1: enable write.	0x0	RW
6	SHT_EN	IDAC1 shutdown enable. Enables automatic shutdown in case of overtemperature. 0: disable this function. 1: enable this function.	0x0	RW
[5:2]	BW	IDAC1 bandwidth. See the IDAC Output Filter section for more details.	0x0	RW
1	PUL	IDAC1 pull down. 0: disable the pull-down current source. 1: enable the pull-down current source.	0x0	RW
0	PD	IDAC1 power down. 0: powers up IDAC1. 1: powers down IDAC1.	0x1	RW

IDAC2 Data Register

Address: 0x40086810, Reset: 0x00000000, Name: IDAC2DAT

Table 32. Bit Descriptions for IDAC2DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC2 high data.	0x0	RW
[16:12]	DATL	IDAC2 low data.	0x0	RW
[11:6]	RESERVED	Reserved.	0x0	R
[5:0]	SYNC	IDAC2 synchronization bits. These six bits are common to the six IDACs. Each bit set to 1 prevents the corresponding channel from updating immediately. The channel updates when the bit changes to 0.	0x00	RW

IDAC2 Control Register

Address: 0x40086814, Reset: 0x01, Name: IDAC2CON

Table 33. Bit Descriptions for IDAC2CON

Bits	Bit Name	Description	Reset	Access
7	CLR	IDAC2 clear bit. 0: clear IDAC1DAT. 1: enable write.	0x0	RW
6	SHT_EN	IDAC2 shutdown enable. Enables automatic shutdown in case of overtemperature. 0: disable this function. 1: enable this function.	0x0	RW
[5:2]	BW	IDAC2 bandwidth. See the IDAC Output Filter section for more details.	0x0	RW
1	PUL	IDAC2 pull down. 0: disable the pull-down current source. 1: enable the pull-down current source.	0x0	RW
0	PD	IDAC2 power down. 0: powers up IDAC2. 1: powers down IDAC2.	0x1	RW

IDAC3 Data Register

Address: 0x40086818, Reset: 0x00000000, Name: IDAC3DAT

Table 34. Bit Descriptions for IDAC3DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC3 high data.	0x0	RW
[16:12]	DATL	IDAC3 low data.	0x0	RW
[11:6]	RESERVED	Reserved.	0x0	R
[5:0]	SYNC	IDAC3 synchronization bits. These six bits are common to the six IDACs. Each bit set to 1 prevents the corresponding channel from updating immediately. The channel updates when the bit changes to 0.	0x00	RW

IDAC3 Control Register

Address: 0x4008681C, Reset: 0x01, Name: IDAC3CON

Table 35. Bit Descriptions for IDAC3CON

Bits	Bit Name	Description	Reset	Access
7	CLR	IDAC3 clear bit. 0: clear IDAC1DAT. 1: enable write.	0x0	RW
6	SHT_EN	IDAC3 shutdown enable. Enables automatic shutdown in case of overtemperature. 0: disable this function. 1: enable this function.	0x0	RW
[5:2]	BW	IDAC3 bandwidth. See the IDAC Output Filter section for more details.	0x0	RW
1	PUL	IDAC3 pull down. 0: disable the pull-down current source. 1: enable the pull-down current source.	0x0	RW
0	PD	IDAC3 power down. 0: powers up IDAC3. 1: powers down IDAC3.	0x1	RW

IDAC4 Data Register

Address: 0x40086820, Reset: 0x0000000000, Name: IDAC4DAT

Table 36. Bit Descriptions for IDAC4DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC4 high data.	0x0	RW
[16:12]	DATL	IDAC4 low data.	0x0	RW
[11:6]	RESERVED	Reserved.	0x0	R
[5:0]	SYNC	IDAC4 sync bits. These six bits are common to the six IDACs. Each bit set to 1 prevents the corresponding channel from updating immediately. The channel updates when the bit changes to 0.	0x00	RW

IDAC4 Control Register

Address: 0x40086824, Reset: 0x01, Name: IDAC4CON

Table 37. Bit Descriptions for IDAC4CON

Bits	Bit Name	Description	Reset	Access
7	CLR	IDAC4 clear bit. 0: clear IDAC1DAT. 1: enable write.	0x0	RW
6	SHT_EN	IDAC4 shutdown enable. Enables automatic shutdown in case of overtemperature. 0: disable this function. 1: enable this function.	0x0	RW
[5:2]	BW	IDAC4 bandwidth. See the IDAC Output Filter section for more details.	0x0	RW
1	PUL	IDAC4 pull down. 0: disable the pull-down current source. 1: enable the pull-down current source.	0x0	RW
0	PD	IDAC4 power down. 0: powers up IDAC4. 1: powers down IDAC4.	0x1	RW

IDAC5 Data Register

Address: 0x40086828, Reset: 0x00000000, Name: IDAC5DAT

Table 38. Bit Descriptions for IDAC5DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC5 high data.	0x0	RW
[16:12]	DATL	IDAC5 low data.	0x0	RW
[11:6]	RESERVED	Reserved.	0x0	R
[5:0]	SYNC	IDAC5 synchronization bits. These six bits are common to the six IDACs. Each bit set to 1 prevents the corresponding channel from updating immediately. The channel updates when the bit changes to 0.	0x00	RW

IDAC5 Control Register

Address: 0x4008682C, Reset: 0x01, Name: IDAC5CON

Table 39. Bit Descriptions for IDAC5CON

Bits	Bit Name	Description	Reset	Access
7	CLR	IDAC5 clear bit. 0: clear IDAC1DAT. 1: enable write.	0x0	RW
6	SHT_EN	IDAC5 shutdown enable. Enables automatic shutdown in case of overtemperature. 0: disable this function. 1: enable this function.	0x0	RW
[5:2]	BW	IDAC5 bandwidth. See the IDAC Output Filter section for more details.	0x0	RW
1	PUL	IDAC5 pull down. 0: disable the pull-down current source. 1: enable the pull-down current source.	0x0	RW
0	PD	IDAC5 power down. 0: powers up IDAC5. 1: powers down IDAC5.	0x1	RW

IDAC6 Data Register

Address: 0x40086830, Reset: 0x00000000, Name: IDAC6DAT

Table 40. Bit Descriptions for IDAC6DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:17]	DATH	IDAC6 high data.	0x0	RW
[16:6]	RESERVED	Reserved.	0x0	R
[5:0]	RESERVED	Reserved. Write 0.	0x0	R

IDAC6 Control Register

Address: 0x40086834, Reset: 0x01, Name: IDAC6CON

Table 41. Bit Descriptions for IDAC6CON

Bits	Bit Name	Description	Reset	Access
7	CLR	IDAC6 Clear bit. 0: clear IDAC6DAT. 1: enable write.	0x0	RW
[6:1]	RESERVED	Reserved.	0x00	R
0	PD	IDAC6 power down. 1: powers IDAC6 off. 0: powers IDAC6 on.	0x1	RW

Table 42. HVA Register Summary

Address	Name	Description	Reset	Access
0x4008E000	HVCON	High voltage control	0x0000	RW

High Voltage Control Register

Address: 0x4008E000, Reset: 0x0000, Name: HVCON

Table 43. Bit Descriptions for HVCON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	OTI_STA	Overtemperature status. 0: normal temperature. 1: overtemperature; high voltage die temperature exceeds temperature setting.		
11	CLAMP_SEL	Enables trimmed SOA shutdown clamp voltage. 1: clamp voltage for SOA shutdown is set to adjust SOA pull-down voltage lowest negative level. 0: trimmed negative pull-down voltage on SOA is disabled. When 0, the lowest voltage is −1.75 V. Default is 0x0, meaning pull-down voltage is approximately −1.75 V.	0x0	RW
[10:9]	OTI_TRIM	These bits control the high voltage die over temperature indicator trip points (OTI). 00: OTI trips at 110°C, typical (junction temperature). 01: OTI trips at 125°C, typical (junction temperature). 10: OTI trips at 130°C, typical (junction temperature). 11: OTI trips at 140°C, typical (junction temperature).	0x0	RW
[8:3]	RESERVED	Reserved.	0x0	RW
2	HV_PD	Powers down high voltage circuitry. 0: normal. 1: power down.	0x0	RW
1	SOA_SINK	Switches on the SOA sink current. 0: switches on the SOA sink current. 1: switches off the SOA sink current.	0x0	RW
0	SOA_VEN	SOA voltage feedback enable. Do not enable when pin is negative. 0: disable SOA voltage feedback to the ADC. 1: enable SOA voltage feedback to the ADC.	0x0	RW

VDACs

VDAC FEATURES

The ADuCM310 has eight VDACs. Four of the VDACs are fully supported on the low voltage analog die. The other four VDACs are supported by the main DAC structure on the low voltage analog die with the output buffer circuits implemented on the high voltage analog die. All eight VDACs are capable of driving a 10 nF load.

VDAC BLOCK DIAGRAM

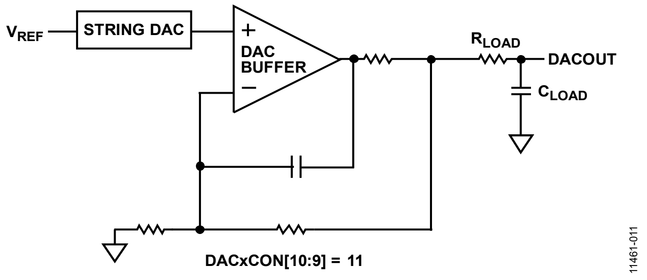


Figure 11. Normal Mode, C_{LOAD} = 10 nF || R_{LOAD} ≥ 75 Ω, 3 V Full Scale (All VDACs)

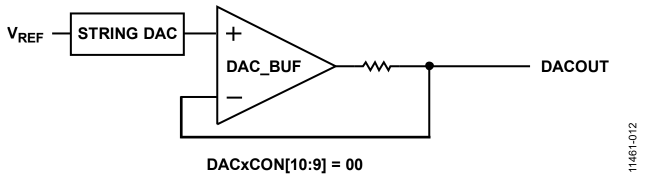


Figure 12. Output Mode Cap Load ≤100 pF (Low Voltage Die Only; VDAC0, VDAC1, VDAC4, and VDAC5)

VDAC OVERVIEW

The ADuCM310 has eight VDACs. Four of the VDACs are fully supported on the low voltage analog die. The other four VDACs are supported by the main DAC structure on the low voltage analog die with the output buffer circuits implemented on the high voltage analog die. All eight VDACs are capable of driving a 10 nF load.

- VDAC0/VDAC1: 0 V to 3 V full-scale output, specified to drive a 75 Ω load, 40 mA maximum. Only the low voltage die is required. These VDACs can select from two reference sources:
  - 0 V to internal reference, V_{REF} (0 V to 2.5 V)
  - 0 V to AV_{DD} (3.3 V)
  - When DAC0CON and DAC1CON[10:9] = 11b, a gain of 1.2 is applied on the DAC output buffer.
- VDAC2/VDAC3: −5 V to 0 V full-scale output, specified to drive a 500 Ω load, 10 mA maximum. Low voltage die and high voltage die required to fully support. The ADuCM310 provides a gain of −2.25. Note that the absolute maximum AV_{NEG} voltage is −5.5 V.
- VDAC4/VDAC5: 0 V to 3 V full-scale output, specified to drive a 300 Ω load, 10 mA maximum. Only the low voltage die is required. These VDACs can select from two reference sources:
  - 0 V to internal reference, V_{REF} (0 V to 2.5 V)
  - 0 V to AV_{DD} (3.3 V)
  - When DAC4CON and DAC5CON[10:9] = 11b, a gain of 1.2 is applied on the DAC output buffer.
- VDAC6: 0 V to 5 V full-scale output, specified to drive a 500 Ω load, 10 mA maximum. Low voltage die and high voltage die required to fully support.
- VDAC7: 0 V to 5 V full-scale output, specified to drive a 100 Ω load, 50 mA maximum. Low voltage die and high voltage die required to fully support.

VDAC OPERATION

The DAC is configurable through a control register and a data register. The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier, as shown in Figure 11 and Figure 12.

VDAC Channel 0, Channel 1, Channel 4, and Channel 5

These four VDAC channels are fully implemented on the low voltage analog die. They are designed to drive a resistive loads of <300 Ω on VDAC4 and VDAC5 and as low as 75 Ω on VDAC0 and VDAC1.

When sourcing high currents (>1 mA), set DACxCON[10] to 1, which results in a smaller maximum output voltage value.

When driving a large capacitive load (>100 pF), set DACxCON [9] to 1. All four of these VDACs are capable of driving a 10 nF capacitive load.

If driving a small capacitive load, <100 pF, and the source current from the VDAC is <1 mA, clear DACxCON[10:9] to 0. This results in a wider output voltage range and reduces the power consumption of the output buffer stages of each of these VDACs. DACxCON[10:9] control the switches that enable/disable the feedback circuitry on the output buffer shown in Figure 1 and Figure 2.

When DACxCON [10:9] = 11, the output voltage range is smaller. See the ADuCM310 data sheet specifications for more details on the output voltage range for each VDAC. The data sheet specifications assume DACxCON [10:9] = 11.

The linearity specification of the DAC when driving a 5 kΩ resistive load to ground is guaranteed through the full transfer function except for Code 0 to Code 100 and, in 0 V to AV_{DD} mode only, Code 3995 to Code 4095. Linearity degradation near ground and AV_{DD} is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is shown in Figure 13.

The dotted line in Figure 13 indicates the ideal transfer function. The solid line represents what the transfer function may look like with endpoint nonlinearities due to saturation of the output amplifier. Figure 13 represents a transfer function in 0 V to AV_{DD} mode only. In 0 V to V_{REF} mode, the lower nonlinearity is similar. However, the upper portion of the transfer function follows the ideal line all the way to the end showing no signs of endpoint linearity errors.

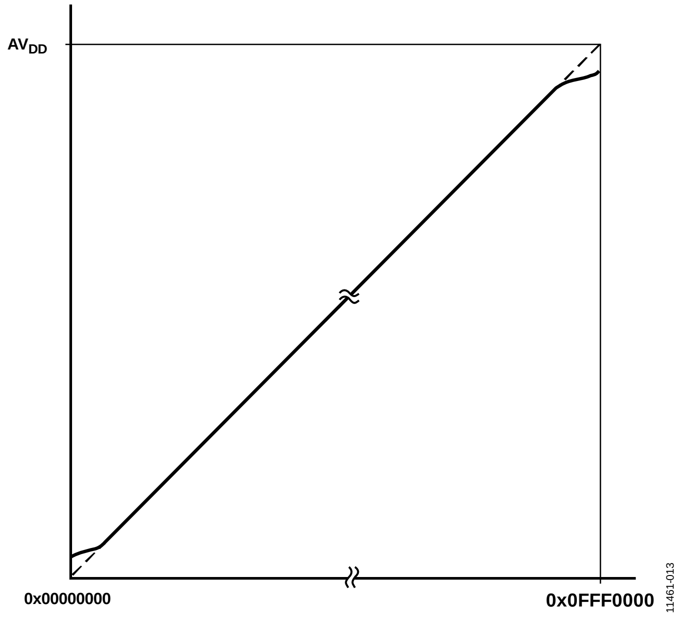


Figure 13. DAC Endpoint Nonlinearities Due to Amplifier Saturation

VDAC Channel 2, Channel 3, Channel 6, and Channel 7

These four VDAC channels are implemented on the low voltage analog die with the output buffer implemented on the high voltage die. The high voltage die implements the amplification and output buffer structure.

The circuitry on the high voltage die for each VDAC assumes an input range of 0 V to 2.5 V. For VDAC 6 and VDAC7, this is then amplified by a factor of 2 to give an output range of 0 V to 5 V. For VDAC 2 and VDAC3, it is amplified and level shifted to 0 V to −5 V. There are no separate memory mapped registers for the high voltage die portion of these VDACs.

Figure 14 shows the headroom limits for VDAC7 when $VDACV_{DD}$ = 5 V for varying output loads. Headroom means the difference between $VDACV_{DD}$ and the maximum allowed output voltage on VDAC7.

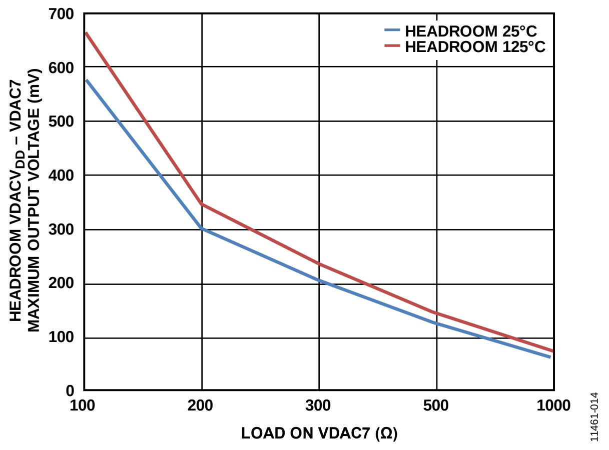


Figure 14. VDAC Headroom Requirements

VDAC Calibration Tables

Offset values are available for all eight VDACs.

For offset, the ATE production test program logs the measured voltage in nV (integer format) to a flash table. Code 0 is applied to all VDACs for this test.

Only use the offset calibration value when setting the DAC output voltage to <150 mV.

Table 44. VDAC Offset Values

VDAC	Calibration Type	Address	Unit	Type	DAC Code Used
0	Offset	0x40C00	nV	Integer	0x00000000
1	Offset	0x40C04	nV	Integer	0x00000000
2	Offset	0x40C08	nV	Integer	0x00000000
3	Offset	0x40C0C	nV	Integer	0x00000000
4	Offset	0x40C10	nV	Integer	0x00000000
5	Offset	0x40C14	nV	Integer	0x00000000
6	Offset	0x40C18	nV	Integer	0x00000000
7	Offset	0x40C1C	nV	Integer	0x00000000

REGISTER SUMMARY: VDAC

Table 45. VDAC Register Summary

Address	Name	Description	Reset	Access
0x40082400	DAC0CON	DAC0 control register	0x0100	RW
0x40082404	DAC1CON	DAC1 control register	0x0100	RW
0x40082408	DAC2CON	DAC2 control register	0x0100	RW
0x4008240C	DAC3CON	DAC3 control register	0x0100	RW
0x40082410	DAC4CON	DAC4 control register	0x0100	RW
0x40082414	DAC5CON	DAC5 control register	0x0100	RW
0x40082418	DAC6CON	DAC6 control register	0x0100	RW
0x4008241C	DAC7CON	DAC7 control register	0x0100	RW
0x40086404	DAC0DAT	DAC0 data register	0x00000000	RW
0x40086408	DAC1DAT	DAC1 data register	0x00000000	RW
0x4008640C	DAC2DAT	DAC2 data register	0x00000000	RW
0x40086410	DAC3DAT	DAC3 data register	0x00000000	RW
0x40086414	DAC4DAT	DAC4 data register	0x00000000	RW
0x40086418	DAC5DAT	DAC5 data register	0x00000000	RW
0x4008641C	DAC6DAT	DAC6 data register	0x00000000	RW
0x40086420	DAC7DAT	DAC7 data register	0x00000000	RW

REGISTER DETAILS: VDAC

DAC0 Control Register

Address: 0x40082400, Reset: 0x0100, Name: DAC0CON

Table 46. Bit Descriptions for DAC0CON

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R
10	DAC0_DRV	DAC0 increased drive. 0: normal drive. 1: for 75 Ω load.	0x0	RW
9	DAC0_10N	DAC0 high load. 0: normal load. 1: can drive 10 nF and full scale = 3 V.	0x0	RW
8	DAC0_PD	DAC0 power down. 0: DAC0 is powered up. 1: DAC0 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC0_EN	DAC0 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC0_RN	DAC0 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC1 Control Register

Address: 0x40082404, Reset: 0x0100, Name: DAC1CON

Table 47. Bit Descriptions for DAC1CON

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R
10	DAC1_DRV	DAC1 increased drive. 0: normal drive. 1: for 75 Ω load.	0x0	RW
9	DAC1_10N	DAC1 high load. 0: normal load. 1: can drive 10 nF and full scale = 3 V.	0x0	RW
8	DAC1_PD	DAC1 power down. 0: DAC1 is powered up. 1: DAC1 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC1_EN	DAC1 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC1_RN	DAC1 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC2 Control Register

Address: 0x40082408, Reset: 0x0100, Name: DAC2CON

Table 48. Bit Descriptions for DAC2CON

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	DAC2_PD	DAC2 power down. 0: DAC2 is powered up. 1: DAC2 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC2_EN	DAC2 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC2_RN	DAC2 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC3 Control Register

Address: 0x4008240C, Reset: 0x0100, Name: DAC3CON

Table 49. Bit Descriptions for DAC3CON

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	DAC3_PD	DAC3 power down. 0: DAC3 is powered up. 1: DAC3 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC3_EN	DAC3 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC3_RN	DAC3 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC4 Control Register

Address: 0x40082410, Reset: 0x0100, Name: DAC4CON

Table 50. Bit Descriptions for DAC4CON

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R
10	DAC4_DRV	DAC4 increased drive. 0: normal drive. 1: for 300 Ω load.	0x0	RW
9	DAC4_10N	DAC4 high load. 0: normal load. 1: can drive 10 nF and full scale = 3 V.	0x0	RW
8	DAC4_PD	DAC4 power down. 0: DAC4 is powered up. 1: DAC4 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC4_EN	DAC4 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC4_RN	DAC4 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC5 Control Register

Address: 0x40082414, Reset: 0x0100, Name: DAC5CON

Table 51. Bit Descriptions for DAC5CON

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x0	R
10	DAC5_DRV	DAC5 increased drive. 0: normal drive. 1: for 300 Ω load.	0x0	RW
9	DAC5_10N	DAC5 high load. 0: normal load. 1: can drive 10 nF and full scale = 3 V.	0x0	RW
8	DAC5_PD	DAC5 power down. 0: DAC5 is powered up. 1: DAC5 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC5_EN	DAC5 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC5_RN	DAC5 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC6 Control Register

Address: 0x40082418, Reset: 0x0100, Name: DAC6CON

Table 52. Bit Descriptions for DAC6CON

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	DAC6_PD	DAC6 power down. 0: DAC6 is powered up. 1: DAC6 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC6_EN	DAC6 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC6_RN	DAC6 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC7 Control Register

Address: 0x4008241C, Reset: 0x0100, Name: DAC7CON

Table 53. Bit Descriptions for DAC7CON

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	DAC7_PD	DAC7 power down. 0: DAC7 is powered up. 1: DAC7 is powered down and output is floating.	0x1	RW
[7:5]	RESERVED	Reserved.	0x0	RW
4	DAC7_EN	DAC7 enable. Must be set to 1. 0: DAC disable. Clear DAC data immediately. 1: DAC enable.	0x0	RW
[3:2]	RESERVED	Reserved.	0x0	RW
[1:0]	DAC7_RN	DAC7 reference selection. These bits set the DAC range. A write to these bits has immediate effect on the DAC. 00: internal reference. 01: reserved. 10: reserved. 11: AV_{DD}/AGND.	0x0	RW

DAC0 Data Register

Address: 0x40086404, Reset: 0x00000000, Name: DAC0DAT

Table 54. Bit Descriptions for DAC0DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC0_DAT	DAC0 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC1 Data Register

Address: 0x40086408, Reset: 0x00000000, Name: DAC1DAT

Table 55. Bit Descriptions for DAC1DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC1_DAT	DAC1 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC2 Data Register

Address: 0x4008640C, Reset: 0x00000000, Name: DAC2DAT

Table 56. Bit Descriptions for DAC2DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC2_DAT	DAC2 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC3 Data Register

Address: 0x40086410, Reset: 0x00000000, Name: DAC3DAT

Table 57. Bit Descriptions for DAC3DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC3_DAT	DAC3 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC4 Data Register

Address: 0x40086414, Reset: 0x00000000, Name: DAC4DAT

Table 58. Bit Descriptions for DAC4DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC4_DAT	DAC4 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC5 Data Register

Address: 0x40086418, Reset: 0x00000000, Name: DAC5DAT

Table 59. Bit Descriptions for DAC5DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC5_DAT	DAC5 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC6 Data Register

Address: 0x4008641C, Reset: 0x00000000, Name: DAC6DAT

Table 60. Bit Descriptions for DAC6DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC6_DAT	DAC6 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

DAC7 Data Register

Address: 0x40086420, Reset: 0x00000000, Name: DAC7DAT

Table 61. Bit Descriptions for DAC7DAT

Bits	Bit Name	Description	Reset	Access
[31:28]	RESERVED	Reserved. Write 0.	0x0	R
[27:16]	DAC7_DAT	DAC7 data.	0x0	RW
[15:0]	RESERVED	Reserved. Write 0.	0x0	R

SYSTEM EXCEPTIONS AND PERIPHERAL INTERRUPTS

CORTEX-M3 AND FAULT MANAGEMENT

The ADuCM310 integrates an ARM Cortex-M3 processor, which supports a number of system exceptions and interrupts generated by peripherals. Table 62 lists the ARM Cortex-M3 processor system exceptions.

Table 62. System Exceptions

Number	Type	Priority	Description
1	Reset	−3 (highest)	Any reset.
2	NMI	−2	Nonmaskable not connected on ADuCM310.
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, data abort, and other address/memory related faults.
6	Usage fault	Programmable	Same as undefined instruction executed or illegal state transition attempt.
7 to 10	Reserved	Not applicable	Reserved.
11	SVCall	Programmable	System service call with SVC instruction. Used for system function calls.
12	Debug monitor	Programmable	Debug monitor (breakpoint, watchpoint, or external debug requests).
13	Reserved	Not applicable	Reserved.
14	PendSV	Programmable	Pendable request for system service. Used for queuing system calls until other tasks and interrupts are serviced.
15	SYSTICK	Programmable	System tick timer.

The peripheral interrupts are controlled by the NVIC and are listed in Table 63. All interrupt sources can wake up the device from Mode 1. Only a limited number of interrupts can wake up the processor from the low power modes (Mode 2 or Mode 3), as shown in Table 63. When the device is woken up from Mode 2 or Mode 3, it returns to Mode 0. If the processor enters any power mode from Mode 1 to Mode 3 while the processor is in an interrupt handler, only an interrupt source with a higher priority than the current interrupt can wake up the device (higher value in IPRx registers).

Two steps are normally required to configure an interrupt:

- Configuring a peripheral to generate an interrupt request to the NVIC.
- Configuring the NVIC for that peripheral request.

Table 63. Interrupt Vector Table

Position Number	Vector	Wake Up Processor from Mode 1	Wake Up Processor from Mode 2 or Mode 3
0	Wake-up timer	Yes	Yes
1	External Interrupt 0	Yes	Yes
2	External Interrupt 1	Yes	Yes
3	External Interrupt 2	Yes	Yes
4	External Interrupt 3	Yes	Yes
5	External Interrupt 4	Yes	Yes
6	External Interrupt 5	Yes	Yes
7	External Interrupt 6	Yes	Yes
8	External Interrupt 7	Yes	Yes
9	External Interrupt 8	Yes	Yes
10	Watchdog timer	Yes	Yes
11	Reserved		
12	Reserved		
13	Low Voltage Die Interrupt 0	Yes	No
14	Reserved		
15	GP Timer 0	Yes	No
16	GP Timer 1	Yes	No
17	Flash controller	Yes	No
18	UART	Yes	No
19	SPI0	Yes	No

Position Number	Vector	Wake Up Processor from Mode 1	Wake Up Processor from Mode 2 or Mode 3
20	SPI1	Yes	No
21	I^{2}C0 slave	Yes	No
22	I^{2}C0 master	Yes	No
23	PLA 0	Yes	No
24	PLA 1	Yes	No
25	DMA error	Yes	No
26	DMA Channel 0 (SPI0 Tx) done	Yes	No
27	DMA Channel 1 (SPI0 Rx) done	Yes	No
28	DMA Channel 2 (SPI1 Tx) done	Yes	No
29	DMA Channel 3 (SPI1 Rx) done	Yes	No
30	DMA Channel 4 (UART Tx) done	Yes	No
31	DMA Channel 5 (UART Rx) done	Yes	No
32	DMA Channel 6 (I^{2}C0 slave Tx) done	Yes	No
33	DMA Channel 7 (I^{2}C0 slave Rx) done	Yes	No
34	DMA Channel 8 (I^{2}C0 master) done	Yes	No
35	DMA Channel 9 (I^{2}C1 slave Tx) done	Yes	No
36	DMA Channel 10 (I^{2}C1 slave Rx) done	Yes	No
37	DMA Channel 11 (I^{2}C1 master) done	Yes	No
38	DMA Channel 12 (ADC) done	Yes	No
39	DMA Channel 13 (Flash) done	Yes	No
40	Reserved		
41	Reserved		
42	Reserved		
43	Reserved		
44	I^{2}C1 slave	Yes	No
45	I^{2}C1 master	Yes	No
46	PLA 2	Yes	No
47	PLA 3	Yes	No
48	GP Timer 2	Yes	No
49	Low Voltage Die Interrupt 1	Yes	No
50	PWM trip	Yes	No
51	PWM PAIR0	Yes	No
52	PWM PAIR1	Yes	No
53	PWM PAIR2	Yes	No
54	PWM PAIR3	Yes	No

Internally to the ARM Cortex-M3 processor, the highest user-programmable priority (0) is treated as fourth priority—after a reset, an NMI, and a hard fault. The ADuCM310 implements three priority bits, which means that eight priority levels are available as programmable priorities. Note that 0 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) determines the order in which the processor activates them. For example, if both SPI0 and SPI1 are Priority Level 1, then SPI0 has higher priority.

To enable an interrupt for any peripheral listed from 0 to 31 in Table 63, set the appropriate bit in the ISER0 register; ISER0 is a 32-bit register and each bit corresponds to the first 32 entries in Table 63.

For example, to enable External Interrupt 3, the interrupt source in the NVIC, set ISER0[4] = 1. Similarly, to disable External Interrupt 3, set ICER0[4] = 1.

To enable an interrupt for any peripheral listed from 32 to 54 in Table 63, set the appropriate bit in the ISER1 register; ISER1 is a 32-bit register and ISER1 Bit 0 to Bit 21 correspond to the entries 32 to 54 in Table 63.

For example, to enable the PWM PAIR0 interrupt source in the NVIC, set ISER1[20] = 1. Similarly, to disable the PWM PAIR0 interrupt, set ICER1[20] = 1.

Alternatively, CMSIS provides a number of useful NVIC functions in the **core_cm3.h** file. The NVIC_EnableIRQ(PWM_PAIR0_IRQn) function enables the PWM PAIR0 interrupt. The interrupt can be disabled by calling the NVIC_DisableIRQ(PWM_PAIR0_IRQn) function.

To set the priority of a peripheral interrupt, the IPRx bits can be set appropriately or, alternatively, the NVIC_SetPriority() function can be called. For example, NVIC_SetPriority(TIMER0_IRQn, 2) configures the GP Timer 0 interrupt with a priority level of 2.

Table 64 lists the registers to enable and disable relevant interrupts and set the priority levels. The registers in Table 64 are defined in the CMSIS **core_cm3.h** file, which is shipped with tools from third party vendors.

Table 64. NVIC Registers

Address	Analog Devices Header File Name	Description	Access
0xE000E004	ICTR	Shows the number of interrupt lines that the NVIC supports.	R
0xE000E010	STCSR	SYSTICK control and status register.	RW
0xE000E014	STRVR	SYSTICK reload value register.	RW
0xE000E018	STCVR	SYSTICK current value register.	RW
0xE000E01C	STCR	SYSTICK calibration value register.	R
0xE000E100	ISER0	Set IRQ0 to IRQ31 enable. Each bit corresponds to Interrupt 0 to Interrupt 31 in Table 63.	RW
0xE000E104	ISER1	Set IRQ32 to IRQ54 enable. Each bit corresponds to interrupt 32 to Interrupt 54 in Table 63.	RW
0xE000E180	ICER0	Clear IRQ0 to IRQ31 by setting the appropriate bit. Each bit corresponds to Interrupt 0 to Interrupt 31 in Table 63	RW
0xE000E184	ICER1	Clear IRQ32 to IRQ54 by setting the appropriate bit. Each bit corresponds to Interrupt 32 to Interrupt 54 in Table 63.	RW
0xE000E200	ISPR0	Set IRQ0 to IRQ31 pending. Each bit corresponds to Interrupt 32 to Interrupt 38 in Table 63.	RW
0xE000E204	ISPR1	Set IRQ32 to IRQ54 pending. Each bit corresponds to Interrupt 32 to Interrupt 54 in Table 63.	RW
0xE000E280	ICPR0	Clear IRQ0 to IRQ31 pending. Each bit corresponds to Interrupt 32 to Interrupt 38 in Table 63.	RW
0xE000E284	ICPR1	Clear IRQ32 to IRQ54 pending. Each bit corresponds to Interrupt 32 to Interrupt 54 in Table 63.	RW
0xE000E300	IABR0	IRQ0 to IRQ31 active bits.	RW
0xE000E304	IABR1	IRQ32 to IRQ54 active bits.	RW
0xE000E400	IPR0	IRQ0 to IRQ3 priority.	RW
0xE000E404	IPR1	IRQ4 to IRQ7 priority.	RW
0xE000E408	IPR2	IRQ8 to IRQ11 priority.	RW
0xE000E40C	IPR3	IRQ12 to IRQ15 priority.	RW
0xE000E410	IPR4	IRQ16 to IRQ19 priority.	RW
0xE000E414	IPR5	IRQ20 to IRQ23 priority.	RW
0xE000E418	IPR6	IRQ24 to IRQ27 priority.	RW
0xE000E41C	IPR7	IRQ28 to IRQ31 priority.	RW
0xE000E420	IPR8	IRQ32 to IRQ35 priority.	RW
0xE000E424	IPR9	IRQ36 to IRQ39 priority.	RW
0xE000E428	IPR10	IRQ40 to IRQ43 priority.	RW
0xE000E42C	IPR11	IRQ44 to IRQ47 priority.	RW
0xE000E430	IPR12	IRQ48 to IRQ51 priority.	RW
0xE000E434	IPR13	IRQ52 to IRQ54 priority.	RW
0xE000ED00	CPUID	CPUID base register.	R
0xE000ED04	ICSR	Interrupt control and status register.	RW
0xE000ED08	VTOR	Vector table offset register.	RW
0xE000ED0C	AIRCR	Application interrupt/reset control register.	RW
0xE000ED10	SCR	System control register.	RW
0xE000ED14	CCR	Configuration control register.	RW
0xE000ED18	SHPR1	System Handlers Register 1.	RW
0xE000ED1C	SHPR2	System Handlers Register 2.	RW
0xE000ED20	SHPR3	System Handlers Register 3.	RW
0xE000ED24	SHCRS	System handler control and state.	RW
0xE000ED28	CFSR	Configurable fault status.	RW

Address	Analog Devices Header File Name	Description	Access
0xE000ED2C	HFSR	Hard fault status.	RW
0xE000ED34	MMAR	Memory manage fault address register.	RW
0xE000ED38	BFAR	Bus fault address.	RW
0xE000EF00	STIR	Software trigger interrupt register.	W

EXTERNAL INTERRUPT CONFIGURATION

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

- Edge: rising edge, falling edge, or both rising and falling edges. An interrupt signal (pulse) is sent to the NVIC upon detecting a transition from low to high, high to low, or on either high to low or low to high.
- Level: high or low. An interrupt signal is generated and remains asserted in the NVIC until the conditions generating the interrupt deassert. The level must be maintained for a minimum of one core clock cycle to be detected.

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

Ensure that the associated GPxIE register bit is enabled for the required external interrupt input. The GPxIE register enables the input path circuit for the external interrupt. It may also be required to enable the internal pull-up resistor if no external pull-up resistor is provided.

For example, for External Interrupt 0, the following code configures P0.3 as an input, enables the pull-up resistor on P0.3, and enables the input path. The appended code also enables external interrupt 0 NVIC interrupt source:

```
pADI_GP0->GPIE = 0x8; // Enable Input path for P0.3 input
pADI_GP0->GPPUL = 0x08; // enable P0.3 pull-up resistor
pADI_INTERRUPT->EI0CFG |= 0x8; // External IRQ0 enabled
NVIC_EnableIRQ(EINT0_IRQn);    // Enable External interrupt 0 source
```

REGISTER SUMMARY: EXTERNAL INTERRUPTS

Table 65. External Interrupts Register Summary

Address	Name	Description	Reset	Access
0x40002420	EI0CFG	External Interrupt Configuration 0	0x0000	RW
0x40002424	EI1CFG	External Interrupt Configuration 1	0x0000	RW
0x40002428	EI2CFG	External Interrupt Configuration 2	0x0000	RW
0x40002430	EICLR	External interrupt clear	0x0000	RW

REGISTER DETAILS: EXTERNAL INTERRUPTS

External Interrupt Configuration Register 0

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

Table 66. Bit Descriptions for EI0CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ3EN	External Interrupt 3 enable bit. 0: External Interrupt 3 disabled. 1: External Interrupt 3 enabled.	0x0	RW
[14:12]	IRQ3MDE	External Interrupt 3 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
11	IRQ2EN	External Interrupt 2 enable bit. 0: External Interrupt 2 disabled. 1: External Interrupt 2 enabled.	0x0	RW
[10:8]	IRQ2MDE	External Interrupt 2 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
7	IRQ1EN	External Interrupt 1 enable bit. 0: External Interrupt 0 disabled. 1: External Interrupt 0 enabled.	0x0	RW
[6:4]	IRQ1MDE	External Interrupt 1 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
3	IRQOEN	External Interrupt 0 enable bit. 0: External Interrupt 0 disabled. 1: External Interrupt 0 enabled.	0x0	RW
[2:0]	IRQ0MDE	External Interrupt 0 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

External Interrupt Configuration Register 1

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

Table 67. Bit Descriptions for EI1CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ7EN	External Interrupt 7 enable bit. 0: External Interrupt 7 disabled. 1: External Interrupt 7 enabled.	0x0	RW
[14:12]	IRQ7MDE	External Interrupt 7 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
11	IRQ6EN	External Interrupt 6 enable bit. 0: External Interrupt 6 disabled. 1: External Interrupt 6 enabled.	0x0	RW
[10:8]	IRQ6MDE	External Interrupt 6 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
7	IRQ5EN	External Interrupt 5 enable bit. 0: External Interrupt 5 disabled. 1: External Interrupt 5 enabled.	0x0	RW
[6:4]	IRQ5MDE	External Interrupt 5 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW
3	IRQ4EN	External Interrupt 4 enable bit. 0: External Interrupt 4 disabled. 1: External Interrupt 4 enabled.	0x0	RW
[2:0]	IRQ4MDE	External Interrupt 4 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

External Interrupt Configuration Register 2

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

Table 68. Bit Descriptions for EI2CFG

Bits	Bit Name	Description	Reset	Access
[15:4]	Reserved	Reserved.	0x0	
3	IRQ8EN	External Interrupt 8 enable bit. 0: External Interrupt 8 disabled. 1: External Interrupt 8 enabled.	0x0	RW
[2:0]	IRQ8MDE	External Interrupt 8 mode registers. 000: rising edge. 001: falling edge. 010: rising or falling edge. 011: high level. 100: low level. 101: falling edge (same as 001). 110: rising or falling edge (same as 010). 111: high level (same as 011).	0x0	RW

External Interrupt Clear Register

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

Table 69. Bit Descriptions for EICLR

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

LOW VOLTAGE ANALOG DIE INTERRUPT CONFIGURATION

Two interrupt lines are available between the low voltage analog die and the interrupt controller on the digital die.

These two interrupt lines are the outputs of two multiplexers of multiple interrupt sources from the low voltage analog die.

The full list of interrupt sources from the low voltage analog die are as follows:

- ADC software conversion complete interrupt. This is asserted at the end of an ADC conversion when this interrupt source is enabled.
- ADC sequencer complete interrupt. This is the interrupt asserted by the ADC sequencer.
- Digital comparator interrupt. If the ADC result is outside the selected threshold, this interrupt is asserted.
- IDAC thermal shutdown interrupt.
- IDAC external reference resistor status interrupt.
- Read ECC interrupt source. Error correction and checking is available on the interface between the digital and analog die. If a read error occurs (for example, an error on the ADC result), this interrupt is asserted.
- Write ECC interrupt source. If the ECC returns an error on a value written to the low voltage die, this interrupt is asserted.

Low Voltage Die Interrupt 1 is more flexible than Low Voltage Interrupt 0. The key differences are as follows:

- Low Voltage Die Interrupt 1 allows all seven different interrupt sources as configured by INTSEL[7:0] to be enabled. In the interrupt handler, the LV1 interrupt source can be determined by the INTSTA register.
- Low Voltage Die Interrupt 0 allows only one of the possible seven interrupt sources selected by INTSEL[15:8] to be enabled at a given time. The INTSTA register is not valid for Low Voltage Interrupt 0.
- To clear an interrupt, set the appropriate bit in the INTCLR register.
- Note there is a delay period required after writing to INTCLR before the associated status bit in the INTSTA register is updated.
- If polling is used of the INTSTA register, the following example code can be used:

```
pADI_LV_INT->INTCLR = 0x1;              // clear Irq source
delay(10);
ucLVIrqStatus = pADI_LV_INT->INTSTA;

// Simple delay routine
void delay (long int length)
{
   while (length >0)
      length--;
}
```

REGISTER SUMMARY: LOW VOLTAGE DIE INTERRUPTS

Table 70. Low Voltage Die Interrupts Register Summary

Address	Name	Description	Reset	Access
0x40083004	INTCLR	Interrupt clear register	0x0000	W
0x40083008	INTSEL	Interrupt mask register	0x0000	RW
0x4008300C	INTSTA	Interrupt status register	0x0000	R

REGISTER DETAILS: LOW VOLTAGE DIE INTERRUPTS

Interrupt Clear Register

Address: 0x40083004, Reset: 0x0000, Name: INTCLR

Table 71. Bit Descriptions for INTCLR

Bits	Bit Name	Description	Reset	Access
7	CLR_WRECC_ERR	Write 1 to this bit to clear the write ECC error interrupt flag.	0x0	W
6	CLR_RDECC_ERR	Write 1 to this bit to clear the read ECC error interrupt flag.	0x0	W
5	CLR_IDAC_EXTRESLOW	Write 1 to this bit to clear the IDAC EXTRESLOW interrupt flag.	0x0	W
4	CLR_IDAC_TSHUT	Write 1 to this bit to clear the IDAC TSHUT interrupt flag.	0x0	W
3	RESERVED	Reserved.	0x0	W
2	CLR_DCOMP	Write 1 to this bit to clear the digital compare interrupt flag.	0x0	W
1	CLR_ADC_SEQ	Write 1 to this bit to clear the ADC sequence conversion interrupt flag.	0x0	W
0	CLR_ADC_SOFTCONV	Write 1 to this bit to clear the ADC software conversion interrupt flag.	0x0	W

Interrupt Mask Register

Address: 0x40083008, Reset: 0x0000, Name: INTSEL

Table 72. Bit Descriptions for INTSEL

Bits	Bit Name	Description	Reset	Access
15	SEL_WRECC_ERR_0	Write 1 to this bit to enable write ECC error interrupt for Interrupt Pin 0.	0x0	RW
14	SEL_RDECC_ERR_0	Write 1 to this bit to enable read ECC error interrupt for Interrupt Pin 0.	0x0	RW
13	SLE_IDAC_EXTRESLOW_0	Write 1 to this bit to enable IDAC EXTRESLOW interrupt for Interrupt Pin 0.	0x0	RW
12	SEL_IDAC_TSHUT_0	Write 1 to this bit to enable IDAC TSHUT interrupt for Interrupt Pin 0.	0x0	RW
11	RESERVED	Reserved.	0x0	RW
10	SEL_DCOMP_0	Write 1 to this bit to enable digital comparator interrupt for Interrupt Pin 0.	0x0	RW
9	SEL_ADC_SEQ_0	Write 1 to this bit to enable ADC sequence conversion interrupt for Interrupt Pin 0.	0x0	RW
8	SEL_ADC_SOFTCONV_0	Write 1 to this bit to enable ADC software conversion interrupt for Interrupt Pin 0.	0x0	RW
7	SEL_WRECC_ERR_1	Write 1 to this bit to enable write ECC error interrupt for Interrupt Pin 1.	0x0	RW
6	SEL_RDECC_ERR_1	Write 1 to this bit to enable read ECC error interrupt for Interrupt Pin 1.	0x0	RW
5	SLE_IDAC_EXTRESLOW_1	Write 1 to this bit to enable IDAC EXTRESLOW interrupt for Interrupt Pin 1.	0x0	RW
4	SEL_IDAC_TSHUT_1	Write 1 to this bit to enable IDAC TSHUT interrupt for Interrupt Pin 1.	0x0	RW
3	RESERVED	Reserved.	0x0	RW
2	SEL_DCOMP_1	Write 1 to this bit to enable digital comparator interrupt for Interrupt Pin 1.	0x0	RW
1	SEL_ADC_SEQ_1	Write 1 to this bit to enable ADC sequence conversion interrupt for Interrupt Pin 1.	0x0	RW
0	SEL_ADC_SOFTCONV_1	Write 1 to this bit to enable ADC software conversion interrupt for Interrupt Pin 1.	0x0	RW

Interrupt Status Register

Address: 0x4008300C, Reset: 0x0000, Name: INTSTA

Table 73. Bit Descriptions for INTSTA

Bits	Bit Name	Description	Reset	Access
7	WRECC_ERR	Write data ECC error interrupt status	0x0	R
6	RDECC_ERR	Read data ECC error interrupt status	0x0	R
5	IDAC_EXTRESLOW	IDAC EXTRESLOW interrupt status	0x0	R
4	IDAC_TSHUT	IDAC temperature TSHT interrupt status	0x0	R
3	RESERVED	Reserved	0x0	R
2	DCOMP	Digital comparator interrupt status	0x0	R
1	ADC_SEQ	ADC sequence interrupt status	0x0	R
0	ADC_SOFTCONV	ADC software conversion interrupt status	0x0	R

RESET

RESET FEATURES

There are four following kinds of resets:

- External reset
- Power-on reset
- Watchdog timeout
- Software system reset

RESET OPERATION

The software system reset is provided as part of the Cortex-M3 processor. To generate a software system reset, the NVIC_SystemReset() function must be called. This effectively writes 0x05FA to the top 16 bits of an AIRCR NVIC register. This function along with other useful functions are defined in the CMSIS header files that are shipped with the tools from third party vendors. The NVIC_SystemReset() function is defined in the **core_cm3.h** file.

The analog peripherals have the option of maintaining their state after a software or watchdog reset. This function is disabled by default. It can be enabled using the LVRST register. Note that while debugging, the software tools generally only issue a software reset, meaning an external reset is needed to return registers to their default values if the retain functionality is enabled.

The GPIO pins and PLA also have the option of maintaining their state after a software or watchdog reset. By default, this function is disabled. Writing a value of 0x0 to RSTCFG configures the GPIO pins and PLA maintain their state after a software or watchdog reset. Before writing to this register, 0x2009 must be written to RSTKEY followed by 0x0426. After the two keys are written to RSTKEY, RSTCFG must be immediately written.

The RSTSTA register stores the cause for the reset until it is cleared by writing the RSTSTA register. RSTSTA can be used during a reset exception service routine to identify the source of the reset.

The watchdog timer is enabled by default after a reset. The default timeout period is approximately 32 seconds.

User code must disable the watchdog timer at the start of user code when debugging or if the watchdog timer is not required.

```
pADI_WDT->T3CON = 0x00 ;                         // Disable watchdog timer
```

Table 74. Device Reset Implications

	Impact					
Reset	**Reset External Pins to Default State**	**Execute Kernel**	**Reset All MMRs Except RSTSTA**	**Reset All Peripherals**	**Valid SRAM**	**RSTSTA After Reset Event**
Software Reset	Yes/No[1]	Yes	Yes/No[1]	Yes/No[1]	Yes/No[2]	RSTSTA[3] = 1
Watchdog Timeout	Yes/No[1]	Yes	Yes/No[1]	Yes/No[1]	Yes/No[2]	RSTSTA[2] = 1
External Reset Pin	Yes	Yes	Yes	Yes	Yes/No[2]	RSTSTA[1] = 1
POR	Yes	Yes	Yes	Yes	No	RSTSTA[0] = 1

[1] GPIO pins, PLA, and analog peripherals have the option of retaining their state during a watchdog or software reset.
[2] RAM is not valid in the case of a reset following a UART download.

REGISTER SUMMARY: RESET

Table 75. Reset Register Summary

Address	Name	Description	Reset	Access
0x40002408	RSTCFG	Reset configuration	0x0000	RW
0x4000240C	RSTKEY	Key protection for RSTCFG	0x0000	RW
0x40002440	RSTSTA	Reset status	0x0000	RW
0x40082C34	LVRST	Low voltage die reset configuration	0x0000	RW

REGISTER DETAILS: RESET

Reset Status Register

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

Table 76. Bit Descriptions for RSTSTA

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

Reset Configuration Register

Address: 0x40002408, Reset: 0x0000, Name: RSTCFG

Table 77. Bit Descriptions for RSTCFG

Bits	Bit Name	Description	Reset	Access
1	HVDIE_RETAIN	High voltage die retain status after watchdog or software reset. 0: high voltage die cannot retain status after watchdog or software reset. Ensure that LVRST[0] = 1. 1: high voltage die retain status after watchdog or software reset. Ensure that LVRST[0] = 0.	0x0	RW
0	GPIO_PLA_RETAIN	GPIO/PLA retain their status after watchdog or software reset. 1: GPIO/PLA do not retain status after watchdog or software reset. 0: GPIO/PLA retain status after watchdog or software reset.	0x0	RW

Key Protection for RSTCFG Register

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

Table 78. Bit Descriptions for RSTKEY

Bits	Bit Name	Description	Reset	Access
[15:0]	RSTKEY	Reset configuration key register. The RSTCFG register is key protected. Two writes to the key are necessary to change the value in the RSTCFG register: first 0x2009, then 0x0426. The RSTCFG register must then be written. A write to any other register on the APB bus before writing to RSTCFG returns the protection to the lock state.	0x0	RW

Low Voltage Die Reset Configuration Register

Address: 0x40082C34, Reset: 0x0000, Name: LVRST

Table 79. Bit Descriptions for LVRST

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0	R
0	RETAIN	Low voltage retains status after watchdog and software reset. 0: low voltage die retains status after watchdog or software reset. 1: low voltage die does not retain status after watchdog or software reset.	0x0	RW

DMA CONTROLLER

DMA FEATURES

The DMA features are as follows:

- 14 dedicated and independent DMA channels
- Two programmable priority levels for each DMA channel
    - Each priority level arbitrates using a fixed priority that is determined by the DMA channel number.
    - Channels with lower number have highest priority. For example, SPI0 transmit has the highest priority; next highest is the SPI0 receive.
- Each DMA channel can access a primary and/or alternate channel control structure
- Supports multiple DMA transfer types
    - Memory to memory
    - Memory to peripheral
    - Peripheral to memory

DMA OVERVIEW

Direct memory access (DMA) provides high speed data transfer between peripherals and memory. Data can be moved quickly by DMA without any processor actions, which keeps processor resources free for other operations.

The DMA controller has 14 channels in total. The 14 channels used are dedicated to managing DMA requests from specific peripherals. Channels are assigned as shown in Table 80.

Table 80. DMA Channel Assignment

Channel	Peripheral
0	SPI0 Tx
1	SPI0 Rx
2	SPI1 Tx
3	SPI1 Rx
4	UART Tx
5	UART Rx
6	I^{2}C0 slave Tx
7	I^{2}C0 slave Rx
8	I^{2}C0 master
9	I^{2}C1 slave Tx
10	I^{2}C1 slave Rx
11	I^{2}C1 master
12	ADC
13	Flash

The channels are connected to dedicated hardware DMA requests; a software trigger is also supported on each channel. This configuration is done by software.

Each DMA channel has a programmable priority level: default or high. Within a priority level, arbitration is done using a fixed priority that is determined by the DMA channel number. Channels with lower number have highest priority. For example, SPI0 transmit has the highest priority; next highest is the SPI0 receive.

The DMA controller supports multiple DMA transfer data widths: independent source and destination transfer size (byte, half word, and word). Source/destination addresses must be aligned on the data size.

The DMA controller supports peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers and access to flash or SRAM, as source and destination.

DMA OPERATION

The DMA controller performs direct memory transfer by sharing the system bus with the Cortex-M3 processor. The DMA request may stall the processor access to the system bus for some bus cycles when the processor and DMA are targeting the same destination (memory or peripheral).

INTERRUPTS

An interrupt can be produced when a transfer is complete for each DMA channel. Separate interrupt enable bits are available in the NVIC for each of the DMA channels.

The DMA controller fetches channel control data structures located in the SRAM memory to perform data transfers. When enabled to use DMA operation, the DMA-capable peripherals request the DMA controller for transfer. At the end of the programmed number of DMA transfers for a channel, the DMA controller generates an interrupt corresponding to that channel. This interrupt indicates the completion of the DMA transfer.

DMA PRIORITY

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

CHANNEL CONTROL DATA STRUCTURE

Every channel has two control data structures associated with it: primary data structure and an alternate data structure. For simple transfer modes, the DMA controller uses either the primary or the alternate data structure. For more complex data transfer modes, such as ping-pong or scatter-gather, the DMA controller uses both the primary and alternate data structures. Each control data structure (primary or alternate) occupies four 32-bit locations in the memory, as shown in Table 81. The entire channel control data structure is shown in Table 82.

Table 81. Channel Control Data Structure

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

Before the controller can perform a DMA transfer, the data structure related to the DMA channel must be programmed at the designated location in system memory, SRAM.

- The source end pointer memory location contains the end address of the source data.
- The destination end pointer memory location contains the end address of the destination data.
- The control data configuration memory location contains the channel configuration control data.

The programming determines the source and destination data size, the number of transfers, and the number of arbitrations.

Table 82. Memory Map of Primary and Alternate DMA Structures

Channel	Primary Structures		Alternate Structures	
Channel 13	Reserved; set to 0	0x0DC	Reserved; set to 0	0x1DC
	Control	0x0D8	Control	0x1D8
	Destination end pointer	0x0D4	Destination end pointer	0x1D4
	Source end pointer	0x0D0	Source end pointer	0x1D0
…	…	…	…	…
Channel 1	Reserved; set to 0	0x01C	Reserved; set to 0	0x11C
	Control	0x018	Control	0x118
	Destination end pointer	0x014	Destination end pointer	0x114
	Source end pointer	0x010	Source end pointer	0x110
Channel 0	Reserved; set to 0	0x00C	Reserved; set to 0	0x10C
	Control	0x008	Control	0x108
	Destination end pointer	0x004	Destination end pointer	0x104
	Source end pointer	0x000	Source end pointer	0x100

The user must define DMA structures in their source code as shown in the examples in the Example Code: Define DMA Structures section. After the structure has been defined, its start address must be assigned to the DMA base address pointer register, DMAPDBPTR.

Each register for each DMA channel is then at the offset address, as specified in Table 82, plus the value in the DMAPDBPTR register.

Example Code: Define DMA Structures

```
memset(dmaChanDesc,0x0,sizeof(dmaChanDesc));
                                 // Setup the DMA base address pointer register.
uiBasPtr = (unsigned int)&dmaChanDesc;         // Setup the DMA base pointer.
pADI_DMA->DMACFG = 1;                          // Enable DMA controller
pADI_DMA->DMAPDBPTR = uiBasPtr;
```

CONTROL DATA CONFIGURATION

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

Table 83. Control Data Configuration

Bits	Name	Description		
[31:30]	DST_INC	Destination address increment. The address increment depends on the source data width as follows:		
		Source Data Width	**DST_INC**	**Destination Address Increment**
		Byte	00	Byte.
			01	Half word.
			10	Word.
			11	No increment. Address remains set to the value that the DST_END_PTR memory location contains.
		Half Word	00	Reserved.
			01	Half word.
			10	Word.
			11	No increment. Address remains set to the value that the DST_END_PTR memory location contains.
		Word	00	Reserved.
			01	Reserved.
			10	Word.
			11	No increment. Address remains set to the value that the DST_END_PTR memory location contains.
[29:28]	DST_SIZE	Size of the destination data. Must match SRC_SIZE. 00: byte. 01: half word. 10: word. 11: reserved.		
[27:26]	SRC_INC	Source address increment. The address increment depends on the source data width as follows:		
		Source Data Width	**DST_INC**	**Source Address Increment**
		Byte	00	Byte.
			01	Half word.
			10	Word.
			11	No increment. Address remains set to the value that the SRC_END_PTR memory location contains.
		Half Word	00	Reserved.
			01	Half word.
			10	Word.
			11	No increment. Address remains set to the value that the SRC_END_PTR memory location contains.
		Word	00	Reserved.
			01	Reserved.
			10	Word.
			11	No increment. Address remains set to the value that the SRC_END_PTR memory location contains.

Bits	Name	Description
[25:24]	SRC_SIZE	Size of the source data. 00: byte. 01: half word. 10: word. 11: reserved.
[23:18]	Reserved	Undefined. Write as 0.
[17:14]	R_POWER	Set these bits to control how many DMA transfers can occur before the controller rearbitrates. Must be set to 0000 for all DMA transfers involving peripherals. Note that the operation of the DMA is indeterminate if a value other than 0000 is programmed in this location for DMA transfers involving peripherals.
[13:4]	N_MINUS_1	The number of configured transfers minus 1 for that channel. The 10-bit value indicates the number of DMA transfers (not the total number of bytes) minus one. The possible values are as follows: 0x000: 1 DMA transfer. 0x001: 2 DMA transfers. 0x002: 3 DMA transfers. … 0x3FF: 1024 DMA transfers.
3	Reserved	Undefined. Write as 0.
[2:0]	CYCLE_CTRL	The transfer types of the DMA cycle. 000: stop (invalid). 001: basic. 010: autorequest. 011: ping-pong. 100: memory scatter-gather primary. 101: memory scatter-gather alternate. 110: peripheral scatter-gather primary. 111: peripheral scatter-gather alternate.

During the DMA transfer process, but before arbitration, CHNL_CFG is written back to system memory with the N_MINUS_1 field changed to reflect the number of transfers yet to be completed.

When the whole DMA cycle is complete, the CYCLE_CTRL bits are made invalid to indicate the completion of the transfer.

DMA TRANSFER TYPES (CHNL_CFG[2:0])

The DMA controller supports five types of DMA transfers. The various types are selected by programming the appropriate values into the CYCLE_CTRL bits (Bits[2:0]) in the CHNL_CFG location of the control data structure.

Invalid (CHNL_CFG[2:0] = 000)

In this mode, no DMA transfer is enabled for the channel. After the controller completes a DMA cycle, it sets the cycle type to invalid to prevent it from repeating the same DMA cycle.

Basic (CHNL_ CFG[2:0] = 001)

In basic mode, the controller can be configured to use either the primary or alternate data structure. The peripheral must present a request for every data transfer. After the channel is enabled, when the controller receives a request, it performs the following operations:

1. The controller performs a transfer. If the number of transfers remaining is zero, the flow continues at Step 3.
2. The controller arbitrates.
    a. If a higher priority channel is requesting service, the controller services that channel.
    b. If the peripheral or software signals a request to the controller, the controller continues at Step 1.
3. At the end of the transfer, the controller generates the corresponding DMA channel interrupt in the NVIC.

Autorequest (CHNL_CFG[2:0] = 010)

In autorequest mode, it is only necessary for the controller to receive a single request to enable it to complete the entire DMA cycle. This allows a large data transfer to occur without significantly increasing the latency for servicing higher priority requests or requiring multiple requests from the processor or peripheral. This mode is useful for a memory-to-memory copy application.

Autorequest is not suitable for peripheral use, except for the ADC sequencer mode where a number of peripheral operations must be completed.

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

1. The controller performs min(2^{R_POWER}, N) transfers for the channel, where R_POWER is Bits[17:14] of the control data configuration register and N is the number of transfers. If the number of transfers remaining is zero, the flow continues at Step 3.
2. A request for the channel is automatically generated. The controller arbitrates. If the channel has the highest priority, the DMA cycle continues at Step 1.
3. At the end of the transfer, the controller generates an interrupt for the corresponding DMA channel.

Ping-Pong (CHNL_CFG[2:0] = 011)

In ping-pong mode, the controller performs a DMA cycle using one of the data structures and then performs a DMA cycle using the other data structure. The controller continues to switch from primary to alternate to primary until it reads a data structure that is invalid, or until the host processor disables the channel.

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

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

Memory Scatter-Gather (CHNL_CFG[2:0] = 100 or 101)

In memory scatter-gather mode, the controller must be configured to use both the primary and alternate data structures. The controller uses the primary data structure to program the control configuration for alternate data structure. The alternate data structure is used for actual data transfers, which are similar to an autorequest DMA transfer. The controller arbitrates after every primary transfer. The controller only requires one request to complete the entire transfer. This mode is used when performing multiple memory-to-memory copy tasks. The processor can configure all of the tasks simultaneously and does not need to intervene in between each task. The controller generates the corresponding DMA channel interrupt in the NVIC when the entire scatter-gather transaction completes using a basic cycle.

In this mode, the controller receives an initial request and then performs four DMA transfers using the primary data structure to program the control structure of the alternate data structure. After this transfer completes, the controller starts a DMA cycle using the alternate data structure. After the cycle completes, the controller performs another four DMA transfers using the primary data structure. The controller continues to switch from primary to alternate to primary until the processor configures the alternate data structure for a basic cycle or until the DMA reads an invalid data structure.

Table 84 lists the fields of the CHNL_CFG memory location for the primary data structure, which must be programmed with constant values for memory scatter-gather mode.

Table 84. CHNL_CFG for Primary Data Structure in Memory Scatter-Gather Mode, CHNL_CFG[2:0] = 100

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

Peripheral Scatter-Gather (CHNL_CFG[2:0] = 110 or 111)

In peripheral scatter-gather mode, the controller must be configured to use both the primary and alternate data structure. The controller uses the primary data structure to program the control structure of the alternate data structure. The alternate data structure is used for actual data transfers, and each transfer takes place using the alternate data structure with a basic DMA transfer. The controller does not arbitrate after every primary transfer. This mode is used when there are multiple peripheral-to-memory DMA tasks to be performed. The Cortex-M3 can configure all of the tasks simultaneously and does not need to intervene in between each task. This is very similar to memory scatter-gather mode except for the arbitration and request requirements. The controller generates the corresponding DMA channel interrupt in the NVIC when the entire scatter-gather transaction completes using a basic cycle.

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

After this cycle completes, the controller rearbitrates, and if it receives a request from the peripheral that has the highest priority, it performs another four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure without rearbitrating. The controller continues to switch from primary to alternate to primary until the processor configures the alternate data structure for a basic cycle or the DMA reads an invalid data structure.

Table 85 lists the fields of the CHNL_CFG memory location for the primary data structure, which must be programmed with constant values for the peripheral scatter-gather mode.

Table 85. CHNL_CFG For Primary Data Structure in Peripheral Scatter Gather Mode, CHNL_CFG[2:0] = 110

Bits	Name	Description
[31:30]	DST_INC	10: configures the controller to use word increments for the address.
[29:28]	DST_SIZE	10: configures the controller to use word transfers.
[27:26]	SRC_INC	10: configures the controller to use word increments for the address.
[25:24]	SRC_SIZE	10: configures the controller to use word transfers.
[23:18]	Reserved	Undefined. Write as 0.
[17:14]	R_POWER	0010: indicates that the DMA controller performed four transfers without rearbitration.
[13: 4]	N_MINUS_1	Configures the controller to perform N DMA transfers, where N is a multiple of 4.
3	Reserved	Undefined. Write as 0.
[2:0]	CYCLE_CTRL	110: configures the controller to perform a memory scatter-gather DMA cycle.

ADDRESS CALCULATION

The DMA controller calculates the source read address based on the content of SRC_END_PTR, the source address increment setting in CHNL_CFG, and the current value of the N_MINUS_1 (CHNL_CFG[13:4]).

Similarly, the destination write address is calculated based on the content of DST_END_PTR, the destination address increment setting in CHNL_CFG, and the current value of the N_MINUS_1 (CHNL_CFG[13:4]).

Source Read Address = *SRC_END_PTR* − (*N_MINUS_1* << (*SRC_INC*)) for *SRC_INC* = 0, 1, 2

Source Read Address = *SRC_END_PTR* for *SRC_INC* = 3

Destination Write Address = *DST_END_PTR* − (*N_MINUS_1* << (*DST_INC*)) for *DST_INC* = 0, 1, 2

Destination Write Address = *DST_END_PTR* for *DST_INC* = 3

where *N_MINUS_1* is the number of configured transfers minus 1 for that channel.

ABORTING DMA TRANSFERS

It is possible to abort a DMA transfer that is in progress by writing to the bit in the DMAENCLR register corresponding to the channel that must be aborted. Do not set DMACFG to 0 because this can corrupt the DMA structures.

REGISTER SUMMARY: DMA

Table 86. DMA Register Summary

Address	Name	Description	Reset	Access
0x40010000	DMASTA	DMA status	0x000F0000	R
0x40010004	DMACFG	DMA configuration	0x00000000	W
0x40010008	DMAPDBPTR	DMA channel primary control data base pointer	0x00000000	RW
0x4001000C	DMAADBPTR	DMA channel alternate control data base pointer	0x00000100	R
0x40010014	DMASWREQ	DMA channel software request	0x00000000	W
0x40010020	DMARMSKSET	DMA channel request mask set	0x00000000	RW
0x40010024	DMARMSKCLR	DMA channel request mask clear	0x00000000	W
0x40010028	DMAENSET	DMA channel enable set	0x00000000	RW
0x4001002C	DMAENCLR	DMA channel enable clear	0x00000000	W
0x40010030	DMAALTSET	DMA channel primary-alternate set	0x00000000	RW
0x40010034	DMAALTCLR	DMA channel primary-alternate clear	0x00000000	W
0x40010038	DMAPRISET	DMA channel priority set	0x00000000	RW
0x4001003C	DMAPRICLR	DMA channel priority clear	0x00000000	W
0x4001004C	DMAERRCLR	DMA per channel bus error	0x00000000	RW
0x40010800	DMABSSET	DMA channel bytes swap enable set	0x00000000	RW
0x40010804	DMABSCLR	DMA channel bytes swap enable clear	0x00000000	W

REGISTER DETAILS: DMA

DMA Status Register

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

Table 87. Bit Descriptions for DMASTA

Bits	Bit Name	Description	Reset	Access
[31:21]	RESERVED	Reserved.	0x0	R
[20:16]	CHNLSM1	Number of available DMA channels minus 1. With 8 channels available, the register reads back 0x07.	0xF	R
[15:8]	RESERVED	Reserved. Undefined.	0x0	R
[7:4]	STATE	Current state of DMA controller. Provides insight into the operation performed by the DMA at the time this register is read. 0000: idle. 0001: reading channel controller data. 0010: reading source data end pointer. 0011: reading destination end pointer. 0100: reading source data. 0101: writing destination data. 0110: waiting for DMA request to clear. 0111: writing channel controller data. 1000: stalled. 1001: done. 1010: peripheral scatter-gather transition. 1011: undefined. … 1111: undefined.	0x0	R
[3:1]	RESERVED	Reserved. Undefined.	0x0	R
0	MENABLE	Enable status of the controller. 0: controller is disabled. 1: controller is enabled.	0x0	R

DMA Configuration Register

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

Table 88. Bit Descriptions for DMACFG

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

DMA Channel Primary Control Data Base Pointer Register

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

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

Table 89. Bit Descriptions for DMAPDBPTR

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

DMA Channel Alternate Control Data Base Pointer Register

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

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

Table 90. Bit Descriptions for DMAADBPTR

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

DMA Channel Software Request Register

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

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

Table 91. Bit Descriptions for DMASWREQ

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x0	W
[13:0]	CHSWREQ	Generate software request. Set the appropriate bit to generate a software DMA request on the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When written: Bit C = 0: does not create a DMA request for Channel C. Bit C = 1: generates a DMA request for Channel C. These bits are automatically cleared by the hardware after the corresponding software request completes.	0x0	W

DMA Channel Request Mask Set Register

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

Table 92. Bit Descriptions for DMARMSKSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Reads back 0.	0x0	R
[13:0]	CHREQMSET	Mask requests from DMA channels. This register disables DMA requests from peripherals. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to mask the request from the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When read: Bit C = 0: requests are enabled for Channel C. Bit C = 1: requests are disabled for Channel C. When written: Bit C = 0: no effect. Use the DMARMSKCLR register to enable DMA requests. Bit C = 1: disables peripheral associated with Channel C from generating DMA requests.	0x0	RW

DMA Channel Request Mask Clear Register

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

Table 93. Bit Descriptions for DMARMSKCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x0	R
[13:0]	CHREQMCLR	Clear REQ_MASK_SET bits in DMARMSKSET. This register enables DMA requests from peripherals by clearing the mask set in the DMARMSKSET register. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to clear the corresponding REQ_MASK_SET bit in DMARMSKSET register. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When written: Bit C = 0: no effect. Use the DMARMSKSET register to disable DMA requests. Bit C = 1: enables peripheral associated with Channel C to generate DMA requests.	0x0	W

DMA Channel Enable Set Register

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

Table 94. Bit Descriptions for DMAENSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x0	R
[13:0]	CHENSET	Enable DMA channels. This register allows the enabling of DMA channels. Reading the register returns the enable status of the channels. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to enable the corresponding channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When read: Bit C = 0: Channel C is disabled. Bit C = 1: Channel C is enabled. When written: Bit C = 0: no effect. Use the DMAENCLR register to disable the channel. Bit C = 1: enables Channel C.	0x0	RW

DMA Channel Enable Clear Register

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

Table 95. Bit Descriptions for DMAENCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHENCLR	Disable DMA channels. This register allows the disabling of DMA channels. Reading the register returns the enable status of the channels. Each bit of the register represents the corresponding channel number in the DMA controller. (Note that the controller disables a channel automatically, by setting the appropriate bit, when it completes the DMA cycle.) Set the appropriate bit to disable the corresponding channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When written: Bit C = 0: no effect. Use the DMAENSET register to enable the channel. Bit C = 1: disables Channel C.	0x0	W

DMA Channel Primary-Alternate Set Register

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

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

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

Table 96. Bit Descriptions for DMAALTSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHPRIALTSET	Control structure status/select alternate structure. Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When read: Bit C = 0: DMA Channel C is using the primary data structure. Bit C = 1: DMA Channel C is using the alternate data structure. When written: Bit C = 0: no effect. Use the DMAALTCLR register to set Bit C to 0. Bit C = 1: selects the alternate data structure for Channel C.	0x0	RW

DMA Channel Primary-Alternate Clear Register

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

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

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

Table 97. Bit Descriptions for DMAALTCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHPRIALTCLR	Select primary data structure. Set the appropriate bit to select the primary data structure for the corresponding DMA channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When written: Bit C = 0: no effect. Use the DMAALTSET register to select the alternate data structure. Bit C = 1: selects the primary data structure for Channel C.	0x0	W

DMA Channel Priority Set Register

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

Table 98. Bit Descriptions for DMAPRISET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHPRISET	Configure channel for high priority. This register enables the user to configure a DMA channel to use the high priority level. Reading the register returns the status of the channel priority mask. Each bit of the register represents the corresponding channel number in the DMA controller. Returns the channel priority mask status, or sets the channel priority to high. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When read: Bit C = 0: DMA Channel C is using the default priority level. Bit C = 1: DMA Channel C is using a high priority level. When written: Bit C = 0: no effect. Use the DMAPRICLR register to set Channel C to the default priority level. Bit C = 1: Channel C uses the high priority level.	0x0	RW

DMA Channel Priority Clear Register

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

Table 99. Bit Descriptions for DMAPRICLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHPRICLR	Configure channel for default priority level. The DMAPRICLR write-only register enables the user to configure a DMA channel to use the default priority level. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to select the default priority level for the specified DMA channel. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When written: Bit C = 0: no effect. Use the DMAPRISET register to set Channel C to the high priority level. Bit C = 1: Channel C uses the default priority level.	0x0	W

DMA Per Channel Bus Error Register

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

Table 100. Bit Descriptions for DMAERRCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	ERRCLR	Bus error status. This register reads and clears the DMA bus error status. The error status is set if the controller encountered a bus error while performing a transfer or when it reads an invalid descriptor (whose cycle control is 3'b000). If a bus error occurs or invalid cycle control is read on a channel, that channel is automatically disabled by the controller. The other channels are unaffected. Write 1 to clear the bits. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When read: Bit C = 0: no bus error/invalid cycle control occurred. Bit C = 1: a bus error/invalid cycle control is pending. When written: Bit C = 0: no effect. Bit C = 1: bit is cleared.	0x0	RW1C

DMA Channel Bytes Swap Enable Set Register

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

Table 101. Bit Descriptions for DMABSSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHBSWAPSET	Byte swap status. This register configures a DMA channel to use the byte. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When read: Bit C = 0: Channel C byte swap is disabled. Bit C = 1: Channel C byte swap is enabled. When written: Bit C = 0: no effect. Use the DMABSCLR register to disable byte swap on Channel C. Bit C = 1: enables byte swap on Channel C.	0x0	RW

DMA Channel Bytes Swap Enable Clear Register

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

Table 102. Bit Descriptions for DMABSCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved. Undefined.	0x0	R
[13:0]	CHBSWAPCLR	Disable byte swap. The DMABSCLR write-only register enables the user to configure a DMA channel to not use byte swapping and use the default operation. Each bit of the register represents the corresponding channel number in the DMA controller. Bit 0 corresponds to DMA Channel 0, and Bit M − 1 corresponds to DMA Channel M − 1. When written: Bit C = 0: no effect. Use the DMABSSET register to enable byte swap on Channel C. Bit C = 1: disables byte swap on Channel C.	0x0	W

FLASH CONTROLLER

FLASH CONTROLLER FEATURES

The flash controller features are as follows:

- 256 kB Flash/EE memory in 2 blocks of 128 kB each (Flash 0 and Flash 1)
- 4 kB information space, which contains factory code

FLASH CONTROLLER OVERVIEW

The flash controller supports read on one flash block and erase/write operation on the other block. There is peripheral DMA support for flash keyhole-based write. A kernel is present in the information space.

The flash controller supports buffered read, that is, executing code from a 64-bit read while fetching next 64 bits.

The flash controller is a 32-bit interface for MMR access. Flash program and erase timing are controlled via a fixed 16 MHz reference clock.

The keyhole is open for access, command fail, and command complete status bits. A cache is provided to speed up execution.

Commands

The flash controller supports the following commands:

- Write command: 64-bits per write
- Page erase commands
- Mass erase commands for each flash block
- Generation of signatures for single or multiple pages
- Command abort supported (by writing to command MMR or by system interrupt)
- Keys required for running commands such as mass erase and the test commands

Protection, Integrity

The flash controller supports the following protection and integrity features:

- Write/read protection for user space
- Read and write protection for information space
- Ability to lock the SW/JTAG interface
- Automatic signature check of information space on reset
- User signature check of user space and information space
- 8-bit ECC
- 1-bit ECC error correction
- 1-bit ECC errors and 2-bit or greater ECC errors can be configured to generate a flash ECC interrupt or a system exception

FLASH CONTROLLER OPERATION

User Space

The flash blocks (Flash 0 and Flash 1) of 128 kB each are available for user code and data. Generally, this can be considered as a 256 kB block from 0 to 0x3FFFF, except that it is not possible to execute from one flash block while erasing or writing parts of the same block.

The top 24 bytes of user space in each flash block are reserved for a signature, the user write protection pattern, and the user flash failure analysis key (USERFAAKEY).

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

Information Space

The information space of Flash 0 and Flash 1 is located at Address 0x40000 to Address 0x40FFF and is divided up between kernel space, test space, and calibration space. Information space is reserved for use by Analog Devices, Inc. Upon reset, the hardware forces the device to execute from the start of information space to copy calibration and configuration values to appropriate MMRs. When the kernel completes, it passes code execution to the start of user code.

The hardware automatically checks the integrity of the kernel after reset. In the event of a failure, FEESTA[13] is set, and user code cannot run. This bit can only be read via a serial wire read if the serial wire interface is enabled.

The kernel code cannot be accessed by the user. The user can read 16 bytes of Flash 0 information space at Address 0x407E8 to Address 0x407F7. These locations contain MANFID0, MANFID1, and the next eight bytes, which are reserved. MANFID0, MANFID1 contain traceability information to uniquely identify every device sold.

The top two bytes at 0x407F4 identify the silicon version and the kernel revision. The first hexadecimal digit in the two bytes translates to the silicon revision, with 0x1 being the first silicon and each future revision increments that value by 1. The next two hexadecimal digits are the ASCII encoded version of the kernel. Prerelease versions start at Y; after release, this changes to the ASCII character 0 and increments upwards if any changes are necessary. The fourth hexadecimal digit represents the kernel minor revision. This starts at 0xE and is decremented for every minor change to the kernel. For example, 0x159A translates to: 1, which indicates first silicon; 59, which is ASCII code to indicate Kernel Revision Y; and A, which indicates Minor Revision A.

There are also hardware registers that identify the version of each silicon die. For more information, see the Silicon Identification section.

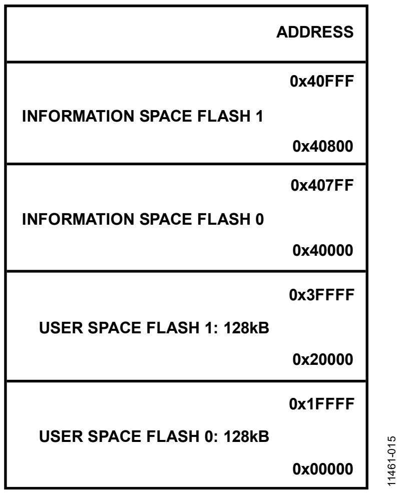

ADDRESS
0x40FFF
INFORMATION SPACE FLASH 1
0x40800
0x407FF
INFORMATION SPACE FLASH 0
0x40000
0x3FFFF
USER SPACE FLASH 1: 128kB
0x20000
0x1FFFF
USER SPACE FLASH 0: 128kB
0x00000
11461-015

Figure 15. Information and User Space Memory Map

Keys

The value 0xF123F456 must be written to the FEEKEY register to run certain user commands, to write to certain locations in flash, or to enable write access to the user setup register (FEECON1).

FLASH MEMORY OPERATION

Keyhole Access

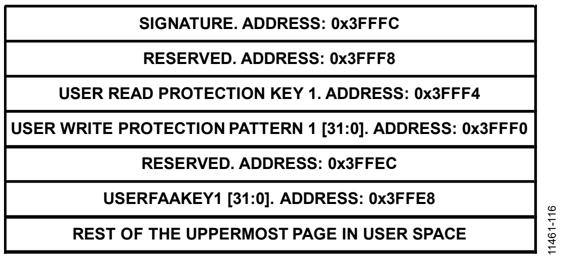
Writing to flash is through keyhole access.

Keyhole access consists of the following:

- Flash address
- Flash data MMR
- Command MMR

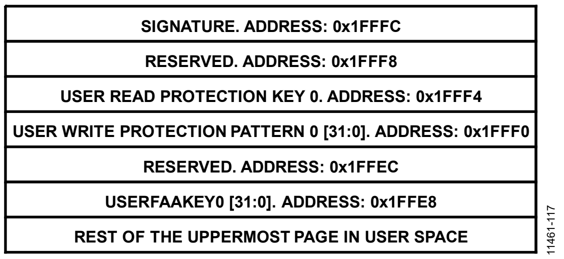
Top of Flash Blocks

The top six words of each flash block have special functionality as listed in Figure 16 and Figure 17. Therefore, normal code or data cannot be placed here.



SIGNATURE. ADDRESS: 0x3FFFC
RESERVED. ADDRESS: 0x3FFF8
USER READ PROTECTION KEY 1. ADDRESS: 0x3FFF4
USER WRITE PROTECTION PATTERN 1 [31:0]. ADDRESS: 0x3FFF0
RESERVED. ADDRESS: 0x3FFEC
USERFAAKEY1 [31:0]. ADDRESS: 0x3FFE8
REST OF THE UPPERMOST PAGE IN USER SPACE

Figure 16. Uppermost Page in User Flash 1 Space



SIGNATURE. ADDRESS: 0x1FFFC
RESERVED. ADDRESS: 0x1FFF8
USER READ PROTECTION KEY 0. ADDRESS: 0x1FFF4
USER WRITE PROTECTION PATTERN 0 [31:0]. ADDRESS: 0x1FFF0
RESERVED. ADDRESS: 0x1FFEC
USERFAAKEY0 [31:0]. ADDRESS: 0x1FFE8
REST OF THE UPPERMOST PAGE IN USER SPACE

Figure 17. Uppermost Page in User Flash 0 Space

Writing to Flash

Each write programs 64 bits of data.

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

1. Write the address of the flash location to FEEFLADR.
2. Write the 64 bits of data to FEEFLDATA0 and FEEFLDATA1.
3. Write the write command to FEECMD.

After the write command is given, the controller writes to flash. CMDDONE (FEESTA[2]) indicates that the command is completed.

Note that a 64-bit location can be written to only once unless it is erased again.

Erasing Flash

User code can call the following three flash erase commands:

- MASSERASE0: this command erases the entire user Flash 0 memory. After entering the user protection key into FEEKEY, write the MASSERASE0 command to FEECMD.
- MASSERASE1: this command erases the entire user Flash 1 memory. After entering the user protection key into FEEKEY, write the MASSERASE1 command to FEECMD.
- PAGEERASE: this command erases 2 kB of flash. The page is selected by FEEADR0. After entering the user protection key into FEEKEY, load FEEADR0 with the page address to be erased. Finally, write the page erase command to FEECMD. CMDDONE (FEESTA[2]) indicates that the command is complete.

During a page or mass erase sequence, the flash controller and flash block consume extra current for the duration of the flash erase sequence.

Signature

The signature checks the integrity of the flash device. The signature is calculated from the lowest 32-bit word to the second highest 32-bit word in the selected block. The signature is a 24-bit CRC with an initial value of 0xFFFFFF and the following polynomial:

$x^{24} + x^{23} + x^6 + x^5 + x + 1$

The data is pushed into the CRC polynomial until the specified end address is reached. A block can be a single page or multiple pages. 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 32 bits are not included when generating the signature. While the signature is being computed for a particular flash, all other accesses to the same flash are stalled.

Note that FEEADR0/FEEADR1 addresses are byte addresses, but only pages must be identified because the lower 11 bits are ignored by the hardware. Also ensure that the addresses written to FEEADR0/FEEADR1 are both either in Flash 0 or Flash 1.

The following code illustrates how the CRC is calculated and how to compare it to the result of the sign command.

```
int FeeCrc(int iLen,int *aiData)
{
int i1,i2,iCrc;
iCrc = 0xffffffff; //Seed value.
for(i1=0; i1<iLen; i1++) //Starting at lowest address.
{
for(i2=31; i2>=0; i2--) //MSB first.
{
iCrc <<= 1; //Left shift.
if((*(aiData+i1))&(1<<i2)) iCrc ^= 0x00800063; //^= Polynomial.
if(iCrc&(1<<24)) iCrc ^= 0x00800063;
}
}
return(iCrc&0x00ffffff); //Return 24 bits.
}
int FeeSign(unsigned long ulStartAddr, unsigned long ulEndAddr)
{
if((pADI_FEE->FEESTA&1)!=0) return 0;
pADI_FEE->FEEADR0 = ulStartAddr;
pADI_FEE->FEEADR1 = ulEndAddr;
pADI_FEE->FEEKEY = 0xF123F456;
pADI_FEE->FEECMD = 0x2;
return 1;
}
FeeSign(0x00800,0x00900); //SIGN for page1.
if(FeeCrc(511,(int *)0x00800) != pADI_FEE->FEESIG)
FlagError();
Else FlagSuccess();
```

ECC Error Handling

During the signature check, the Error Checking and Correcting (ECC) is checked on each 72-bit flash read (64-bit flash read and 8-bit ECC). If errors are corrected by the ECC, the ERRDETECTED flag in the status register, FEESTA, is set after the signature check is completed. If errors are detected and cannot be corrected by ECC, the ERRDETECTED flag in FEESTA is set. A signature check is treated as a failure when the computed signature is not equal to the stored signature.

During a read of the flash, if there is a 1-bit error, the error is corrected by default but neither ECC interrupts nor system exceptions are enabled. If interrupts or system exceptions are not enabled by the user, the appropriate flags in FEESTA are not set in the event of an ECC error.

A 1-bit ECC interrupt or system exception can be enabled in the ECC enable/disable register (FEEECCCONFIG), if required. If the appropriate interrupts or system exceptions are enabled in the FEEECCCONFIG register, the appropriate flags are set in the status register.

If there is a 2-bit ECC error and if interrupts or system exceptions are enabled in the FEEECCCONFIG register, an error is issued by the controller. If the appropriate interrupts or system exceptions are enabled in the FEEECCCONFIG register, the appropriate flags are set in the status register.

An ECC error is signaled by the ECC error detection/correction hardware when a flash location is read. Depending on the location which flash (Flash 0/Flash 1) the read happens, the appropriate flags are set in the status register (ECCREADERRFLSH0, ECCREADERRFLSH1, and so on). Note that 1-bit errors corrected meet full data sheet specifications.

If a system exception is enabled the device vectors to a hard fault or bus fault in the event of an ECC error, see the SHCSR register in the ARM Cortex-M3 processor documentation to enable a bus fault; {see the ARM Cortex-M3 processor documentation for more information on the SHCSR processor.

ECC Error During Read

Two separate ECCREADERR flags are present in the status register: FEESTA[10:9] and FEESTA[12:11] for Flash 0 and Flash 1. If the interrupt is configured to be generated when an ECC error occurs, the address at which the error is detected is available for the user. If a system exception is configured, the BFAR register contains the address for which ECC error is detected.

ECC Error During Execution of Sign Command

If there is an ECC error during the signature check, the registers are not updated. After the command is complete, ECCERRCMD flags in FEESTA[8:7] are updated. No interrupt or system exception is generated.

Flash Protection

The following three types of protection are implemented:

- Key protection
- Read protection
- Write protection

Flash Protection: Key Protection

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

The user key is 0xF123F456. This key must be entered to run certain user commands, write to certain locations in flash or to enable write access to FEECON1. Once entered, the key remains asserted unless a command is written to FEECMD. When the command starts, the key clears automatically. If this key is entered to enable write access to FEECON1 or to enable writes to certain locations in flash, it must be cleared by user code afterwards. To clear the key, write any value other than 0xF123F456 to FEEKEY.

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 Bit 0 of FEECON1. 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 exits to user code, it enables serial wire access unless either of the keys at 0x3FFF4 or 0x1FFF4 is set to 0x0000003A. This means that the device is always read protected after either key is in place and that no debug access can occur.

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 flash controller generates an interrupt if the command error/complete interrupt are enabled. The write protection for each block is stored near the top of each block. The top four bytes are for a signature; the next eight are reserved. The next 32-bit flash location contains the protection pattern which is copied to FEEPRO0 and FEEPRO1 at startup with each bit protecting a block of 4 kB of flash. If no protection is specified, protection can be set by writing to FEEPRO0 and FEEPRO1.

Flash Failure Analysis Key

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

The user must set the key as two 32-bit values near the top of each user flash block. Supplying this key to Analog Devices allows access to user code for debug purposes. See Figure 16 and Figure 17 for details.

Flash Controller Abort

Commands (erase, sign, or mass verify) and writes can be aborted upon receipt of an interrupt as listed in Table 63. Aborts are also possible by writing an abort command to the FEECMD register. However, if flash is being programmed and the routine controlling the programming is in flash, it is not possible to use the abort command to abort the cycle because instructions cannot be read. Therefore, the ability to abort a cycle on the assertion of any system interrupt is provided. The FEEAENx register enables aborts upon receipt of an interrupt. Each bit in the FEEAENx registers corresponds to an interrupt listed in Table 63. Setting a bit in the FEEAENx register enables the corresponding interrupt to abort flash operations.

When a command or write is aborted via a system interrupt, FEESTA[5:4] indicates an abort (FEESTA[5:4] = 11).

Depending on the state that a write cycle is in when the abort asserts, the write cycle may or may not complete. If the write or erase cycle did not complete, a fail status of aborted can be read in the status register.

If an immediate response to an interrupt is required during an erase or program cycle, the interrupt service routine and the interrupt vector table must be moved to SRAM or must be in the other flash block for the duration of the cycle.

If the DMA engine is set up to write a block of data to flash, an interrupt can be set up to abort the current write; however, the DMA engine starts the next write immediately. The interrupt causing the abort stays asserted so that there is a number of aborted write cycles in this case before the processor gains access to flash.

When an abort is triggered by an interrupt, all commands are repeatedly aborted until the appropriate FEEAENx bit is cleared or the interrupt source is cleared.

CPU Execution Speed

The basic execution speed of the ADuCM310 is one CPU cycle per clock cycle. The default clock speed is 80 MHz. This speed is achieved when running from cache but is slightly less when running directly from flash. An average execution speed of over 70 MHz is typically achieved for typical C code. For more details and how to achieve full speed operation for critical code, see the AN-1322 Application Note, *ADuCM320 Code Execution Speed*.

Memory Cache

A memory cache is provided on-chip to speed up program execution. The instruction cache is configured and set up by default. If the user writes code to the flash, the user must perform a chip reset to ensure that old cached data is cleared and that the new code can be executed. If a chip reset is not an option, the following code can clear the cache. iCache must be 0x10001 or (CACHESETUP_IINIT_EN| CACHESETUP_DINIT_EN) to clear both the instruction and data cache.

```
int FeeCacheClr(int iCache)
   {
   unsigned int   ui1;

   ui1 = pADI_FEE->CACHESETUP;
   pADI_FEE->CACHEKEY = 0xf123f456;
   pADI_FEE->CACHESETUP = ui1|(iCache&(CACHESETUP_IINIT_EN|CACHESETUP_DINIT_EN));
   while(pADI_FEE->CACHESTAT&(iCache&(CACHESETUP_IINIT_EN|CACHESETUP_DINIT_EN)));
   pADI_FEE->CACHEKEY = 0xf123f456;
   pADI_FEE->CACHESETUP = ui1;
   return 1;
   }
```

Normally, programming tools used take care of this when downloading code onto devices.

Flash DMA Support

Flash controller operations can be supported by DMA. This feature is software configurable. The two flash blocks are independent, meaning that the user can continue executing from one block while programming another block. The DMA is very useful for this because the core must only initiate the write to flash and the DMA takes care of it in the background, triggering an interrupt when the operation is complete. The following code can be used for writing to flash using the DMA.

```
void FLASHDMAINIT(void)
{
pADI_DMA->DMACFG = 0x1;    // Enable DMA mode in DMA controller
Dma_Init();
NVIC_EnableIRQ(DMA_FLASH_IRQn);  // Enable Flash DMA IRQ
FLASHDMAWRITE(uxFlashData, 64);
pADI_DMA->DMAENSET = 0x2000;
pADI_FEE->FEEFLADR = uiAdr;
pADI_FEE->FEEKEY = 0xF123F456;
pADI_FEE->FEECON1 |= (FEECON1_KHDMA_EN);  // Enable Flash DMA mode
}
void FLASHDMAWRITE (unsigned char * pucTX_DMA, unsigned int iNumVals)
{
DmaDesc Desc;
// Common configuration of all the descriptors used here
Desc.ctrlCfg.Bits.cycle_ctrl = DMA_BASIC;
desc.ctrlcfg.bits.next_useburst = 0x0;
desc.ctrlcfg.bits.r_power = 1;
desc.ctrlcfg.bits.src_prot_ctrl = 0x0;
Desc.ctrlCfg.Bits.dst_prot_ctrl = 0x0;
Desc.ctrlCfg.Bits.src_size = DMA_SIZE_WORD;
Desc.ctrlCfg.Bits.dst_size = DMA_SIZE_WORD;
// TX Primary Descriptor
Desc.srcEndPtr = (unsigned int)(pucTX_DMA+ 4*(iNumVals - 0x1) );
Desc.destEndPtr = (unsigned int)&(pADI_FEE->FEEFLDATA1);
Desc.ctrlCfg.Bits.n_minus_1 = iNumVals - 0x1;
Desc.ctrlCfg.Bits.src_inc = DMA_SRCINC_WORD;
Desc.ctrlCfg.Bits.dst_inc = DMA_DSTINC_NO;
*Dma_GetDescriptor(Flash_C) = Desc;
}

void DMA_Flsh_Int_Handler()
{
   pADI_FEE->FEEKEY = 0xF123F456;
   pADI_FEE->FEECON1 &= (~FEECON1_KHDMA_EN);           // Disable Flash DMA mode
   dma_done = 1;
}
```

Flash Controller Performance and Command Duration

All flash functions are slower than the CPU execution speed. The CPU Execution Speed section details the slight penalty of slower flash reads. All other flash operations are significantly slower, as detailed in Table 103.

Table 103. Typical Flash Execution Times

Operation	Time (Typical)	Comments
Write 64-bit location	75 µs	
Mass erase one flash block	18 ms	
Page erase one page	18 ms	
Sign Flash 0/Flash 1 information space	33 µs	512 cycles, 2 kB
Sign Flash 0/Flash 1 user space	2.1 ms	32,000 cycles, 128 kB

In general, these timings are a guideline only and software must use the flash status information or the interrupt system to detect when flash operations are complete. If one of the operations in Table 103 is executed in the same block as the block from which the CPU fetches instructions, the CPU stalls until the operation is complete.

REGISTER SUMMARY: FLASH CONTROLLER

Table 104. Flash Controller Register Summary

Address	Name	Description	Reset	Access
0x40018000	FEESTA	Status register	0x00000000	R
0x40018004	FEECON0	Command control register: interrupt enable register	0x00000000	RW
0x40018008	FEECMD	Command register	0x00000000	RW
0x4001800C	FEEFLADR	Flash address keyhole register	0x00000000	RW
0x40018010	FEEFLDATA0	Flash data register: keyhole interface lower 32 bits	0x00000000	RW
0x40018014	FEEFLDATA1	Flash data register: keyhole interface upper 32 bits	0x00000000	RW
0x40018018	FEEADR0	Lower page address register	0x00000000	RW
0x4001801C	FEEADR1	Upper page address register	0x00000000	RW
0x40018020	FEEKEY	Key register	0x00000000	W
0x40018028	FEEPRO0	Write protection register for Flash 0	0xFFFFFFFF	RW
0x4001802C	FEEPRO1	Write protection register for Flash 1	0xFFFFFFFF	RW
0x40018034	FEESIG	Upper half word of signature	0x0000000X	R
0x40018038	FEECON1	User setup register	0x0000000X	RW
0x40018040	FEEWRADDRA	Write abort address register	0x0000000X	R
0x40018048	FEEAEN0	Interrupt abort enable register: Interrupt 31 to Interrupt 0	0x00000000	RW
0x4001804C	FEEAEN1	Interrupt abort enable register: Interrupt 54 to Interrupt 32	0x000000	RW
0x40018064	FEEECCCONFIG	ECC enable/disable, error response	0x00000000	RW
0x40018074	FEEECCADDR0	Flash 0 ECC Error Address	0x00000000	R
0x40018078	FEEECCADDR1	Flash 1 ECC Error Address	0x00000000	R
0x400180C0	CACHESTAT	Cache status register	0x2	R
0x400180C4	CACHESETUP	Cache setup register	0x2	RW
0x400180C8	CACHEKEY	Cache key register	0x0	W

REGISTER DETAILS: FLASH CONTROLLER

Status Register

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

Table 105. Bit Descriptions for FEESTA

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x0	R
[28:27]	ECCREADERRIBUS	Instruction bus ECC error during a read of flash if a system exception is enabled. 00: no error (NOERR). Successful read from Flash 1. 01: error detected (ERRDETECTED). 2-bit error detected in one or more flash locations during a read from Flash 1. The errors are not corrected. 10: error corrected (ERRCORRECTED). 1-bit error detected for one flash location during read from Flash 1. The error is corrected. 11: 1-bit and 2-bit error detected (ERR1BIT_2BIT). During the read, 1-bit error and 2-bit errors are detected in Flash 1.	0x0	RC
[26:25]	ECCREADERRDBUS	Data bus ECC error during a read of flash if a system exception is enabled. 00: no error (NOERR). Successful read from Flash 1. 01: error detected (ERRDETECTED). 2-bit error detected in one or more flash locations during a read from Flash 1. The errors are not corrected. 10: error corrected (ERRCORRECTED). 1-bit error detected for one flash location during read from Flash 1. The error is corrected. 11: 1-bit and 2-bit error detected (ERR1BIT_2BIT). During the read, 1-bit error and 2-bit errors are detected in Flash 1.	0x0	RC
[24:22]	ECCCOUNTFLASH1	This is a 3-bit counter that reflects the number of 1-bit ECC read errors in Flash 1 after FEESTA[12:11] = 0x2 and before FEESTA is read. This counter does not count on ECC 2-bit errors. The counter is cleared when FEESTA is read by the user.	0x0	R
[21:20]	RESERVED	Reserved.	0x0	R
[19:17]	ECCCOUNTFLASH 0	This is a 3-bit counter that reflects the number of 1-bit ECC read errors in Flash 0 after FEESTA[10:9] = 0x2 and before FEESTA is read. This counter does not count on ECC 2-bit errors. The counter is cleared when FEESTA is read by the user.	0x0	RC
[16:15]	ECCERRSIGN	ECC error during initial signature check. 00: no error (NOERR). Successful flash read operation during the initial signature check or page signature check. 01: error detected (ERRDETECTED). During the initial signature check, 2-bit errors are detected and not corrected for at least one flash location. 10: error corrected (ERRCORRECTED). 1-bit error is corrected for one flash location during a signature command. 11: 1-bit and 2-bit error detected (ERR1BIT_2BIT). During the initial signature command, 1-bit errors and 2-bit errors are detected on one or more flash locations.	0x0	R

Bits	Bit Name	Description	Reset	Access
13	SIGNERR	Information space signature check on reset error. After a reset, the flash controller automatically checks the information space signature. If the signature check fails, this bit is asserted. The user can check if this bit is set via serial wire only. User code does not execute if this bit is set. The bit is cleared if the correct signature is programmed to the most significant long word in information space.	0x0	R
[12:11]	ECCREADERRFLSH1	ECC errors during a read of Flash 1 if interrupt is enabled. 00: no error (NOERR). Successful read from Flash 1. 01: error detected (ERRDETECTED). 2-bit error detected in one or more flash locations during a read from Flash 1. The errors are not corrected. 10: error corrected (ERRCORRECTED). 1-bit error detected for one flash location during the read from Flash 1. The error is corrected. 11: 1-bit and 2-bit error detected (ERR1BIT_2BIT). During the read, 1-bit error and 2-bit errors are detected in Flash 1.	0x0	RC
[10:9]	ECCREADERRFLSH0	ECC errors during read of Flash 0 if interrupt is enabled. 00: no error (NOERR). Successful read from Flash 0. 01: error detected (ERRDETECTED). 2-bit error detected in one or more flash locations during a read from Flash 0. The errors are not corrected. 10: error corrected (ERRCORRECTED). 1-bit error detected for one flash location while during read from Flash 0. The error is corrected. 11: 1-bit and 2-bit error detected (ERR1BIT_2BIT). During the read, 1-bit error and 2-bit errors are detected in Flash 0.	0x0	RC
[8:7]	ECCERRCMD	ECC errors during signature commands. 00: no error (NOERR). Successful flash read operation during the signature check. 01: error detected (ERRDETECTED). 2-bit error detected in one or more flash locations during the signature command. The errors are not corrected. 10: error corrected (ERRCORRECTED). 1-bit error detected for one flash location while doing a signature check. The error is corrected. 11: 1-bit and 2-bit error detected (ERR1BIT_2BIT). During the signature command, 1-bit error and 2-bit errors are detected on one or more flash locations.	0x0	RC
6	RESERVED	Reserved.	0x0	R
[5:4]	CMDRES	These two bits indicate the status of a command on completion or the status of a write. If multiple commands are executed or there are multiple writes without a read of the status register, the first error encountered is stored. Cleared to 0 when read. 00: successful completion of a command or a write. 01: attempted signature check, write, or erase of a protected location. 10: read verify error. After an erase, the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all Fs, this is the resulting status. If the sign command is executed and the resulting signature does not match the data in the upper 4 bytes of the upper page in a block, this is the resulting status. 11: indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.	0x0	RC
3	WRALMOSTDONE	Write almost complete, keyhole registers open for access. This bit flags the earliest point at which the flash controller data and address may be updated for the next command without affecting an active flash command operation. 0: cleared to 0 when read. 1: set to 1 when a write completes.	0x0	RC
2	CMDDONE	This bit asserts when a command completes. If there are multiple commands, this status bit asserts after the first command completes and stays asserted until read. 0: cleared to 0 when read. 1: set to 1 when a command completes.	0x0	RC

Bits	Bit Name	Description	Reset	Access
1	WRCLOSE	This bit is asserted when the user has written all keyhole registers for flash write, and the controller has started the write. If this bit is high, all keyhole registers (FEEFLADR, FEEFLDATA0, FEEFLDATA1), except the command register (FEECMD), are closed for write.	0x0	R
0	CMDBUSY	Command busy. This bit is asserted when the flash block is executing any command entered via the command register.	0x0	R

Command Control Register: Interrupt Enable Register

Address: 0x40018004, Reset: 0x00000000, Name: FEECON0

Table 106. Bit Descriptions for FEECON0

Bits	Bit Name	Description	Reset	Access
[31:3]	RESERVED	Returns 0 when read.	0x0	R
2	IENERR	Command fail interrupt enable. If this bit is set, an interrupt is generated when a command or flash write completes with an error status. 0: disable. 1: enable.	0x0	RW
1	IWRALCOMP	Write almost complete interrupt enable. Returns 0 when read. 0: disable. 1: enable.	0x0	RW
0	IENCMD	Command complete interrupt enable. When set, an interrupt is generated when a command or flash write completes. 0: disable. 1: enable.	0x0	RW

Command Register

Address: 0x40018008, Reset: 0x00000000, Name: FEECMD

Table 107. Bit Descriptions for FEECMD

Bits	Bit Name	Description	Reset	Access
[31:5]	RESERVED	Returns 0x0. Always returns 0 when read.	0x0	RW
[4:0]	CMD	00000: IDLE. No command executed. 00001: PAGEERASE. Write the address of the page to be erased to the FEEADR0 register, then write this code to the FEECMD, and the flash erases the page. When the erase is complete, the flash reads every location in the page to verify all words in the page are erased. If there is a read verify error, it is indicated in the status register. To erase multiple pages, wait until a previous page erase has completed, check the status, and then issue a command to start the next page erase. Before entering this command, 0xF123F456 must be written to the FEEKEY register. 00010: 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 the FEEADR0 register, write the address of the last page to the FEEADR1 register, then write this code to the FEECMD register. When the command is complete, the signature is available for reading in the sign register. The last 4 bytes of the last page in a block is reserved for storing the signature. Before entering this command, 0xF123F456 must be written to the FEEKEY register. 00100: WRITE. Use this command to write to flash locations. This command requires a user key for writing into write protection location and USERFAAKEY location. No key is required for other flash locations. This command takes the address and data from the FEEADR and FEEFLDATA key-hole registers. 00101: MASSERASE0. Erase all of Flash 0 user space. To enable this operation, 0xF123F456 must be written to the FEEKEY register (this is to prevent accidental erases). When the mass erase is complete, the controller reads every location to verify that all locations are 0xFFFFFFFFFFFFFFFF. If there is a read verify error, it is indicated in the status register. 00110: MASSERASE1. Erase all of Flash 1 user space. To enable this operation, 0xF123F456 must be written to the FEEKEY register (this is to prevent accidental erases). When the mass erase is complete, the controller reads every location to verify that all locations are 0xFFFFFFFFFFFFFFFF. If there is a read verify error, it is indicated in the status register. 01000: ABORT. If this command is issued, any command currently in progress is stopped. The status indicates command completed with an error status (FEESTA[5:4] = 0x3). Note that this is the only command that can be issued while another command is already in progress. This command can also stop a write that may be 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 successfully. To enable this operation, 0xF123F456 must be written to the FEEKEY register (this is to prevent accidental aborts). All other combinations are reserved.	0x0	RW

Flash Address Keyhole Register

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

Table 108. Bit Descriptions for FEEFLADR

Bits	Bit Name	Description	Reset	Access
[31:19]	RESERVED	Returns 0x0 if read.	0x0	R
[18:3]	FLADDR	Memory mapped address for the flash location. Specifies flash address for write command. The 3 LSBs always read 0.	0x0	RW
[2:0]	RESERVED	Returns 0x0 if read.	0x0	R

Flash Data Register: Keyhole Interface Lower 32 Bits

Address: 0x40018010, Reset: 0x00000000, Name: FEEFLDATA0

Table 109. Bit Descriptions for FEEFLDATA0

Bits	Bit Name	Description	Reset	Access
[31:0]	FLDATA0	FLDATA0 forms the lower 32 bit of the 64-bit data to be written to flash.	0x0	RW

Flash Data Register: Keyhole Interface Upper 32 Bits

Address: 0x40018014, Reset: 0x00000000, Name: FEEFLDATA1

Table 110. Bit Descriptions for FEEFLDATA1

Bits	Bit Name	Description	Reset	Access
[31:0]	FLDATA1	FLDATA1 forms the upper 32 bit of the 64-bit data to be written to flash.	0x0	RW

Lower Page Address Register

Address: 0x40018018, Reset: 0x00000000, Name: FEEADR0

Table 111. Bit Descriptions for FEEADR0

Bits	Bit Name	Description	Reset	Access
[31:19]	RESERVED	Return 0 when read.	0x0	RW
[18:11]	PAGEADDR0	Used by sign and page erase commands for specifying page address. See the description of these commands in Table 107.	0x0	RW
[10:0]	RESERVED	Reserved.	0x0	R

Upper Page Address Register

Address: 0x4001801C, Reset: 0x00000000, Name: FEEADR1

Table 112. Bit Descriptions for FEEADR1

Bits	Bit Name	Description	Reset	Access
[31:19]	RESERVED	Return 0 when read.	0x0	RW
[18:11]	PAGEADDR1	Used by sign command for specifying the endpage address. See the description of this command in Table 107.	0x0	RW
[10:0]	RESERVED	Reserved.	0x0	R

Key Register

Address: 0x40018020, Reset: 0x00000000, Name: FEEKEY

Table 113. Bit Descriptions for FEEKEY

Bits	Bit Name	Description	Reset	Access
[31:0]	KEY	Enter 0xF123F456 to allow key protected operations. Returns 0x00 if read.	0x0	W

Write Protection Register For Flash 0

Address: 0x40018028, Reset: 0xFFFFFFFF, Name: FEEPRO0

Table 114. Bit Descriptions for FEEPRO0

Bits	Bit Name	Description	Reset	Access
[31:0]	WRPROT0	Write protection for Flash 0 – 32 bits. Each bit corresponds to a 4 kB flash section. Writing 0 to a bit protects the corresponding section of flash. This register is read-only if the write protection in flash has been programmed.	0xFFFFFFFF	RW

Write Protection Register For Flash 1

Address: 0x4001802C, Reset: 0xFFFFFFFF, Name: FEEPRO1

Table 115. Bit Descriptions for FEEPRO1

Bits	Bit Name	Description	Reset	Access
[31:0]	WRPROT1	Write protection for Flash1 – 32 bits. Each bit corresponds to a 4 kB flash section. Writing 0 to a bit protects the corresponding section of flash. This register is read-only if the write protection in flash has been programmed.	0xFFFFFFFF	RW

Upper Half Word of Signature Register

Address: 0x40018034, Reset: 0x0000000X, Name: FEESIG

Table 116. Bit Descriptions for FEESIG

Bits	Bit Name	Description	Reset	Access
[31:24]	RESERVED	Returns 0x0 if read.	0x0	R
[23:0]	SIGN	24-bit signature.	0xx	R

User Setup Register

Address: 0x40018038, Reset: 0x00000001, Name: FEECON1

This register is key protected; therefore, the key (0xF123F456) must be entered in FEEKEY. After writing to FEECON1, a value other than 0xF123F456 must be written again to FEEKEY to reassert the key protection.

Table 117. Bit Descriptions for FEECON1

Bits	Bit Name	Description	Reset	Access
[31:3]	RESERVED	Returns 0 when read.	0x0	R
2	INCR	Auto-increment FEEFLAADR for non-DMA operation. 0: disable automatic address increment. 1: enable automatic address increment.	0x0	RW
1	KHDMA	Keyhole DMA enable. 0: disable DMA mode. 1: enable DMA mode.	0x0	RW
0	DBG	JTAG debug enable. If this bit is 1, access via the serial wire debug interface is enabled. If this bit is 0, access via the serial wire debug interface is disabled. The kernel set this bit to 1 when it has finished executing, thus enabling debug access to a user. 0: disable JTAG access. 1: enable JTAG access.	0x1	RW

Write Abort Address Register

Address: 0x40018040, Reset: 0x0000000X, Name: FEEWRADDRA

Table 118. Bit Descriptions for FEEWRADDRA

Bits	Bit Name	Description	Reset	Access
[31:0]	WRABORTADDR	If a write is aborted, this register contains the address of the location being written when the write was aborted. This register has appropriate value if command abort happened. This register must be read after the command is aborted and must be read before any other command is given. After reset, the value is random.	0xx	R

Interrupt Abort Enable Register: Interrupt 31 to Interrupt 0

Address: 0x40018048, Reset: 0x00000000, Name: FEEAEN0

Table 119. Bit Descriptions for FEEAEN0

Bits	Bit Name	Description	Reset	Access
[31:0]	SYSIRQABORTEN	Lower 32 bits of system interrupt abort enable. To allow a system interrupt to abort a command (write, erase, sign, or mass verify), write a 1 to the appropriate bit in this register. Each bit corresponds to one interrupt listed in the interrupt vector table.	0x0	RW

Interrupt Abort Enable Register: Interrupt 54 to Interrupt 32

Address: 0x4001804C, Reset: 0x000000, Name: FEEAEN1

Table 120. Bit Descriptions for FEEAEN1

Bits	Bit Name	Description	Reset	Access
[22:0]	SYSIRQABORTEN	Upper 23 bits of system interrupt abort enable. To allow a system interrupt to abort a command (write, erase, sign, or mass verify), write a 1 to the appropriate bit in this register. Each bit corresponds to one interrupt listed in the interrupt vector table.	0x0	RW

ECC Enable/Disable, Error Response Register

Address: 0x40018064, Reset: 0x00000000, Name: FEEECCCONFIG

This register is key protected; the key (0x5ECCACCE) must be entered in FEEKEY. After writing to FEECCCONFIG, the key is cleared.

Table 121. Bit Descriptions for FEEECCCONFIG

Bits	Bit Name	Description	Reset	Access
[31:5]	RESERVED	Reserved	0x0	R
[4:3]	ECCCMDINTEN	Interrupt enabled (flash interrupt) when an ECC error occurs during a read. 00: interrupt is not generated if an ECC error occurs while reading from the flash. 01: interrupt enabled only if a 2-bit error is detected during a read from Flash 0 or Flash 1. 10: interrupt enabled only if a 1-bit error is detected during a read from Flash 0 or Flash 1. 11: interrupt enabled if either a 2-bit error or 1-bit error is detected during a read from Flash 0 or Flash 1.	0x0	RW
[2:1]	ECCCMDAHBEN	Generates a system exception (bus fault) when an ECC error occurs during a read. 00: exception is not generated if an ECC error occurs while reading from flash. 01: exception enabled only if a 2-bit error is detected during a read from Flash 0 or Flash 1. 10: exception enabled only if a 1-bit error is detected during a read from Flash 0 or Flash 1. 11: exception enabled if either a 2-bit error or 1-bit error is detected during a read from Flash 0 or Flash 1.	0x0	RW
0	ECCDISABLE	Setting this bit to 1 disables ECC. When ECC is disabled, the ECC module is bypassed. When a read to a flash location is carried out, corresponding to the requested address, LSB 32 or MSB 32 bit raw-data is returned to the bus.	0x0	RW

Flash 0 ECC Error Address Register

Address: 0x40018074, Reset: 0x00000000, Name: FEEECCADDR0

Table 122. Bit Descriptions for FEEECCADDR0

Bits	Bit Name	Description	Reset	Access
[31:19]	RESERVED	Reserved.	0x0	R
[18:0]	VALUE	This register has the address of Flash 0 for which the ECC error is detected.	0x0	R

Flash 1 ECC Error Address Register

Address: 0x40018078, Reset: 0x00000000, Name: FEEECCADDR1

Table 123. Bit Descriptions for FEEECCADDR1

Bits	Bit Name	Description	Reset	Access
[31:19]	RESERVED	Reserved.	0x0	R
[18:0]	VALUE	This register has the address of Flash 1 for which ECC error is detected.	0x0	R

Cache Status Register

Address: 0x400180C0, Reset: 0x00000002, Name: CACHESTAT

Table 124. Bit Descriptions for CACHESTAT

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	R
18	DLOCK	This bit is set when the data cache is locked and cleared when the data cache is unlocked.	0x0	R
17	DEN	If this bit is set, the data cache is enabled and when cleared, the data cache is disabled. This bit is also cleared when CACHESTAT[16] is set.	0x0	R
16	DINIT	This bit is set when the data cache memory initialization starts and clears when initialization is done. The data cache is disabled when this bit is set.	0x0	R
[15:4]	RESERVED	Reserved.	0x0	R
2	ILOCK	This bit is set when the instruction cache is locked and cleared when the instruction cache is unlocked.	0x0	R
1	IEN	If this bit is set, the instruction cache is enabled and when cleared, the instruction cache is disabled. This bit is also cleared when CACHESTAT[0] is set.	0x1	R
0	IINIT	This bit is set when the instruction cache memory initialization starts and clears when initialization is done. The instruction cache is disabled when this bit is set.	0x0	R

Cache Setup Register

Address: 0x400180C4, Reset: 0x00000002, Name: CACHESETUP

This register is key protected; therefore, the key (0xF123F456) must be entered in CACHEKEY.

Table 125. Bit Descriptions for CACHESETUP

Bits	Bit Name	Description	Reset	Access
[31:20]	RESERVED	Reserved.	0x0	RW
19	DWRBUF	If this bit is set, for every AHB access, a hit from the write buffer is not checked.	0x0	RW
18	DLOCK	If this bit is set, data cache contents are locked. Any new misses are not replaced in data cache. This bit is cleared when CACHESETUP[16] is set.	0x0	RW
17	DEN	If this bit set, D-Cache is enabled for AHB accesses. If 0, data cache is disabled, and all AHB accesses are via Flash memory. This bit is cleared when CACHESETUP[16] is set.	0x0	RW
16	DINIT	If this bit is set, the data cache contents are initialized to all zeros. This bit is cleared once the initialization starts.	0x0	RW
[15:5]	RESERVED	Reserved.	0x0	RW
4	IRDBUF	If this bit is set, for every AHB access, a hit from the read buffer is not checked.	0x0	RW
3	IWRBUF	If this bit is set, for every AHB access, a hit from the write buffer is not checked.	0x0	RW
2	ILOCK	If this bit is set, instruction cache contents are locked. Any new misses are not replaced in instruction cache. This bit is cleared when CACHESETUP[0] is set.	0x0	RW
1	IEN	If this bit set, I-Cache is enabled for AHB accesses. If 0, instruction cache is disabled, and all AHB accesses are via flash memory. This bit is cleared when CACHESETUP[0] is set.	0x1	RW
0	IINIT	If this bit is set, the instruction cache contents are initialized to all zeros. This bit is cleared once the initialization starts.	0x0	RW

Cache Key Register

Address: 0x400180C8, Reset: 0x00000000, Name: CACHEKEY

Table 126. Bit Descriptions for CACHEKEY

Bits	Bit Name	Description	Reset	Access
[31:0]	KEY	Cache key register. Enter 0xF123F456 to allow key protected operations. Returns 0x0 if read. The key is cleared automatically after writing to the setup register.	0x0	W

SILICON IDENTIFICATION

The ADuCM310 has three silicon die, and each die has a register that identifies the silicon.

On the digital die, the CHIPID register contains the silicon version in the bottom 4 bits and, in the following 12 bits, the device identification.

On the high voltage die, the silicon revision is stored in the HVCON register.

REGISTER SUMMARY: SILICON IDENTIFICATION

Table 127. Silicon ID Register Summary

Address	Name	Description	Reset	Access
0x40002024	CHIPID	Digital die ID	0x0562	R
0x40082C30	LVID	Low voltage die ID	0x0074	R

REGISTER DETAILS: SILICON IDENTIFICATION

Digital Die ID Register

Address: 0x40002024, Reset: 0x0562, Name: CHIPID

Table 128. Bit Descriptions for CHIPID

Bits	Bit Name	Description	Reset	Access
[15:4]	PARTID	Device identifier	0x56	R
[3:0]	REV	Silicon revision number	0x2	R

Low Voltage Die ID Register

Address: 0x40082C30, Reset: 0x0074, Name: LVID

Table 129. Bit Descriptions for LVID

Bits	Bit Name	Description	Reset	Access
[15:0]	LVID	LVID comes from hardware coded logic.	0x74	R

DIGITAL INPUTS/OUTPUTS

DIGITAL INPUTS/OUTPUTS FEATURES

The ADuCM310 features a number of bidirectional general-purpose input/output (GPIO) pins. Most of the GPIO pins have multiple functions, configurable by user code. At power up, all but one of these pins are configured as GPIOs; one pin reflects the state of the POR. This pin can also be configured by user code to be used as a GPIO.

DIGITAL INPUTS/OUTPUTS BLOCK DIAGRAM

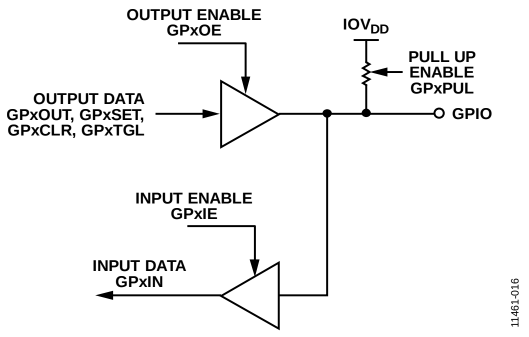


Figure 18. GPIO Structure

DIGITAL INPUTS/OUTPUTS OVERVIEW

The GPIOs are grouped into four ports: Port 0, Port 1, and Port 2 contains eight GPIOs, and Port 3 contains four GPIOs. Each GPIO can be configured as input, output, or fully open circuit and has an internal pull-up programmable resistor with a drive capability of 1 mA. All input/output pins are functional over the full supply range (IOV_{DD} = 2.9 V to 3.6 V (maximum)), and the logic input voltages are specified as percentages of the supply as follows:

$V_{INL} = 0.2 \times IOV_{DD}\ max$

$V_{INH} = 0.7 \times IOV_{DD}\ min$

The absolute maximum input voltage is IOV_{DD} + 0.3 V. The typical leakage current of the GPIOs configured as input or open circuit is 50 nA per GPIO. When the ADuCM310 enters a power saving mode, the GPIO pins retain their states. Note that a driving peripheral cannot drive the pin. That is, if the UART is driving the pin upon entry to deep sleep, it is isolated from the pin and power is gated. Its state and control are restored upon wake up.

DIGITAL INPUTS/OUTPUTS OPERATION

Input/Output Pull-Up Enable

All GPIO pins have an internal pull-up resistor with a drive capability of 3 mA. Using the GPxPUL register, it is possible to enable/disable pull-up registers on the pins when they are configured as inputs. The pull-ups are automatically disabled when the GPIO pin is set as an output or when open circuit is enabled. The pull-ups are also disabled by default.

The pull-up is implemented as a MOS device; therefore, the pull-up resistor value varies with the voltage on the pin.

If a pin is configured as an open-drain output, it is not possible to enable the internal pull-up; an external pull-up is required. This only affects open-drain output mode, not input mode.

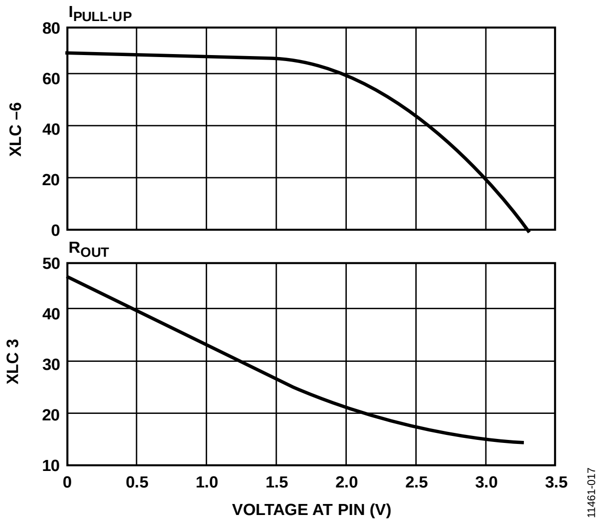


Figure 19. GPIO Pull-Up Resistor Value

Input/Output Data In

When configured as an input (by default), the GPIO input levels are available in the GPxIN register.

Open-Drain Enable

This disables the input paths if the pin is set as an output. To disable the input and not drive the pin, set the open drain and drive Logic 1. External interrupts are not available when open drain is enabled.

If a pin is configured as an open-drain output, it is not possible to enable the internal pull-up; an external pull-up is required. This only affects open-drain output mode, not input mode.

To enable a pin as an open-drain output, set the appropriate bit in the GPxOEN and GPxODE registers.

Input/Output Data Out

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

Bit Set

Bit set mode sets one or more GPIO data outputs without affecting others within a port. Only the GPIO corresponding with the write data bit equal to 1 is set; the remaining GPIOs are unaffected.

Bit Clear

Bit clear mode clears one or more GPIO data outputs without affecting others within a port. Only the GPIO corresponding with the write data bit equal to 1 is cleared; the remaining GPIOs are unaffected.

Bit Toggle

Bit toggle mode toggles one or more GPIO data outputs without affecting others within a port. Only the GPIO corresponding to the write data bit equal to 1 is toggled; the remaining GPIOs are unaffected.

Input/Output Data Output Enable

The data output path is enabled; the values in the GPxOUT register are reflected on the GPIOs.

DIGITAL PORT MULTIPLEX

This block provides control over the GPIO functionality of specified pins because some of the pins have a choice to work as a GPIO or to have other specific functions.

Table 130. GPIO Multiplex Table

	Configuration Modes			
GPIO	**00**	**01**	**10**	**11**
GP0—GP0CON Controls These Bits				
P0.0	GPIO (GP0CON[1:0] = 0x0)	SPI0 SCLK (GP0CON[1:0] = 0x1)		PLAI[0] (GP0CON[1:0] = 0x3)
P0.1	GPIO (GP0CON[3:2] = 0x0)	SPI0 MISO (GP0CON[3:2] = 0x1)		PLAI[1] (GP0CON[3:2] = 0x3)
P0.2	GPIO (GP0CON[5:4] = 0x0)	SPI0 MOSI (GP0CON[5:4] = 0x1)		PLAI[2] (GP0CON[5:4] = 0x3)
P0.3	GPIO/IRQ0 (GP0CON[7:6] = 0x0)	SPI0 CS (GP0CON[7:6] = 0x1)	PLACLK0 (GP0CON[7:6] = 0x2)	PLAI[3] (GP0CON[7:6] = 0x3)
P0.4	GPIO (GP0CON[9:8] = 0x0)	I^{2}C0 SCL (GP0CON[9:8] = 0x1)		PLAO[2] (GP0CON[9:8] = 0x3)
P0.5	GPIO (GP0CON[11:10] = 0x0)	I^{2}C0 SDA (GP0CON[11:10] = 0x1)		PLAO[3] (GP0CON[11:10] = 0x3)
P0.6	GPIO (GP0CON[13:12] = 0x0)	I^{2}C1 SCL (GP0CON[13:12] = 0x1)		PLAO[4] (GP0CON[13:12] = 0x3)
P0.7	GPIO (GP0CON[15:14] = 0x0)	I^{2}C1 SDA (GP0CON[15:14] = 0x1)		PLAO[5] (GP0CON[15:14] = 0x3)
GP1—GP1CON Controls These Bits				
P1.0	GPIO (GP1CON[1:0] = 0x0)	UART SIN (GP1CON[1:0] = 0x1)	ECLKIN (GP1CON[1:0] = 0x2)	PLAI[4] (GP1CON[1:0] = 0x3)
P1.1	GPIO (GP1CON[3:2] = 0x0)	UART SOUT (GP1CON[3:2] = 0x1)	PLACLK1 (GP1CON[3:2] = 0x2)	PLAI[5] (GP1CON[3:2] = 0x3)
P1.2	GPIO (GP1CON[5:4] = 0x0)	PWM0 (GP1CON[5:4] = 0x1)		PLAI[6] (GP1CON[5:4] = 0x3)
P1.3	GPIO (GP1CON[7:6] = 0x0)	PWM1 (GP1CON[7:6] = 0x1)		PLAI[7] (GP1CON[7:6] = 0x3)
P1.4	GPIO (GP1CON[9:8] = 0x0)	PWM2 (GP1CON[9:8] = 0x1)	SPI1 SCLK (GP1CON[9:8] = 0x2)	PLAO[10] (GP1CON[9:8] = 0x3)
P1.5	GPIO (GP1CON[11:10] = 0x0)	PWM3 (GP1CON[11:10] = 0x1)	SPI1 MISO (GP1CON[11:10] = 0x2)	PLAO[11] (GP1CON[11:10] = 0x3)
P1.6	GPIO (GP1CON[13:12] = 0x0)	PWM4 (GP1CON[13:12] = 0x1)	SPI1 MOSI (GP1CON[13:12] = 0x2)	PLAO[12] (GP1CON[13:12] = 0x3)
P1.7	GPIO/IRQ1 (GP1CON[15:14] = 0x0)	PWM5 (GP1CON[15:14] = 0x1)	SPI1 CS (GP1CON[15:14] = 0x2)	PLAO[13] (GP1CON[15:14] = 0x3)

GPIO	Configuration Modes			
	00	**01**	**10**	**11**
GP2—GP2CON Controls These Bits				
P2.0	GPIO/IRQ2 (GP2CON[1:0] = 0x0)	PWMTRIP (GP2CON[1:0] = 0x1)	PLACLK2 (GP2CON[1:0] = 0x2)	PLAI[8] (GP2CON[1:0] = 0x3)
P2.1	GPIO/IRQ3 (GP2CON[3:2] = 0x0)	PWMSYNC (GP2CON[3:2] = 0x1)		PLAI[9] (GP2CON[3:2] = 0x3)
P2.2	GPIO/IRQ4 (GP2CON[5:4] = 0x0)		CLK OUT (GP2CON[5:4] = 0x2)	PLAI[10] (GP2CON[5:4] = 0x3)
P2.3	GPIO/BM (GP2CON[7:6] = 0x0)			
P2.4	GPIO/IRQ5 (GP2CON[9:8] = 0x0)	ADCCONV (GP2CON[9:8] = 0x1)	PWM6 (GP2CON[9:8] = 0x2)	PLAO[18] (GP2CON[9:8] = 0x3)
P2.5	GPIO/IRQ6 (GP2CON[11:10] = 0x0)		PWM7 (GP2CON[11:10] = 0x2)	PLAO[19] (GP2CON[11:10] = 0x3)
P2.6	GPIO/IRQ7 (GP2CON[13:12] = 0x0)			PLAO[20] (GP2CON[13:12] = 0x3)
P2.7	GPIO/IRQ8 (GP2CON[15:12] = 0x0)			PLAO[21] (GP2CON[15:14] = 0x3)
P3.0	GPIO (GP3CON[1:0] = 0x0)			PLAI[12] (GP3CON[1:0] = 0x3)
P3.1	GPIO (GP3CON[3:2] = 0x0)			PLAI[13] (GP3CON[3:2] = 0x3)
P3.2	GPIO (GP3CON[5:4] = 0x0)			PLAI[14] (GP3CON[5:4] = 0x3)
P3.3[1]	GPIO (GP3CON[7:6] = 0x0)			PLAI[15] (GP3CON[7:6] = 0x3)
P3.4	GPIO (GP3CON[9:8] = 0x0)			PLAO[26] (GP3CON[9:8] = 0x3)
P3.5[1]	GPIO (GP3CON[11:10] = 0x0)			PLAO[27] (GP3CON[11:10] = 0x3)
P3.6[1]	GPIO (GP3CON[13:12] = 0x0)			PLAO[28] (GP3CON[13:12] = 0x3)
P3.7[1]	GPIO (GP3CON[15:14] = 0x0)			PLAO[29] (GP3CON[15:14] = 0x3)

[1] Not available as an external pin. Internal PLA elements connected to these pins can be used.

REGISTER SUMMARY: DIGITAL INPUT/OUTPUT

Table 131. GPIO Register Summary

Address	Name	Description	Reset	Access
0x40020000	GP0CON	GPIO Port 0 configuration	0x0000	RW
0x40020004	GP0OEN	GPIO Port 0 output enable	0x00	RW
0x40020008	GP0PUL	GPIO Port 0 pull-up enable	0x00	RW
0x4002000C	GP0IE	GPIO Port 0 input path enable	0xFF	RW
0x40020010	GP0IN	GPIO Port 0 registered data input	0xXX	R
0x40020014	GP0OUT	GPIO Port 0 data output	0x0000	RW
0x40020018	GP0SET	GPIO Port 0 data out set	0x00	W
0x4002001C	GP0CLR	GPIO Port 0 data out clear	0x00	W
0x40020020	GP0TGL	GPIO Port 0 pin toggle	0x00	W
0x40020024	GP0ODE	GPIO Port 0 open-drain enable	0x00	RW
0x40020040	GP1CON	GPIO Port 1 configuration	0x0000	RW
0x40020044	GP1OEN	GPIO Port 1 output enable	0x00	RW
0x40020048	GP1PUL	GPIO Port 1 pull-up enable	0x00	RW
0x4002004C	GP1IE	GPIO Port 1 input path enable	0xFF	RW
0x40020050	GP1IN	GPIO Port 1 registered data input	0xXX	R
0x40020054	GP1OUT	GPIO Port 1 data output	0x0000	RW
0x4002005C	GP1CLR	GPIO Port 1 data out clear	0x00	W
0x40020058	GP1SET	GPIO Port 1 data out set	0x00	W
0x40020060	GP1TGL	GPIO Port 1 pin toggle	0x00	W
0x40020064	GP1ODE	GPIO Port 1 open-drain enable	0x00	RW
0x40020080	GP2CON	GPIO Port 2 configuration	0x0010	RW
0x40020084	GP2OEN	GPIO Port 2 output enable	0x00	RW
0x40020088	GP2PUL	GPIO Port 2 pull-up enable	0x00	RW
0x4002008C	GP2IE	GPIO Port 2 input path enable	0xFF	RW
0x40020090	GP2IN	GPIO Port 2 registered data input	0xXX	R
0x40020094	GP2OUT	GPIO Port 2 data output	0x0000	RW
0x40020098	GP2SET	GPIO Port 2 data out set	0x00	W
0x4002009C	GP2CLR	GPIO Port 2 data out clear	0x00	W
0x400200A0	GP2TGL	GPIO Port 2 pin toggle	0x00	W
0x400200A4	GP2ODE	GPIO Port 2 open-drain enable	0x00	RW
0x400200C0	GP3CON	GPIO Port 3 configuration	0x0000	RW
0x400200C4	GP3OEN	GPIO Port 3 output enable	0x00	RW
0x400200C8	GP3PUL	GPIO Port 3 pull-up enable	0x00	RW
0x400200CC	GP3IE	GPIO Port 3 input path enable	0xFF	RW
0x400200D0	GP3IN	GPIO Port 3 registered data input	0xXX	R
0x400200D4	GP3OUT	GPIO Port 3 data output	0x0000	RW
0x400200D8	GP3SET	GPIO Port 3 data out set	0x00	W
0x400200DC	GP3CLR	GPIO Port 3 data out clear	0x00	W
0x400200E0	GP3TGL	GPIO Port 3 pin toggle	0x00	W
0x400200E4	GP3ODE	GPIO Port 3 open-drain enable	0x00	RW

REGISTER DETAILS: DIGITAL INPUT/OUTPUT

GPIO Port Configuration Registers

Address: 0x40020000, Reset: 0x0000, Name: GP0CON

Address: 0x40020040, Reset: 0x0000, Name: GP1CON

Address: 0x40020080, Reset: 0x0000, Name: GP2CON

Table 132. Bit Descriptions for GP0CON, GP1CON, and GP2CON

Bits	Bit Name	Description[1]	Reset	Access
[15:14]	CON7	Configuration bits for Port x.7. See Table 130.	0x0	RW
[13:12]	CON6	Configuration bits for Port x.6. See Table 130.	0x0	RW
[11:10]	CON5	Configuration bits for Port x.5. See Table 130.	0x0	RW
[9:8]	CON4	Configuration bits for Port x.4. See Table 130.	0x0	RW
[7:6]	CON3	Configuration bits for Port x.3. See Table 130.	0x0	RW
[5:4]	CON2	Configuration bits for Port x.2. See Table 130.	0xx[2]	RW
[3:2]	CON1	Configuration bits for Port x.1. See Table 130.	0x0	RW
[1:0]	CON0	Configuration bits for Port x.0. See Table 130.	0x0	RW

[1] Where x is 0 for Port 0, 1 for Port 1, and 2 for Port 2.
[2] Reset value for Port 0 and Port 1 is 0x0. Reset value for Port 2 is 0x1.

GPIO Port Output Enable Registers

Address: 0x40020004, Reset: 0x00, Name: GP0OEN

Address: 0x40020044, Reset: 0x00, Name: GP1OEN

Address: 0x40020084, Reset: 0x00, Name: GP2OEN

Table 133. Bit Descriptions for GP0OEN, GP1OEN, and GP2OEN

Bits	Bit Name	Description	Reset	Access
[7:0]	OE	Pin output drive enable. 0: disable the output on the corresponding GPIO. 1: enable the output on the corresponding GPIO.	0x00	RW

GPIO Port Pull-Up Enable Registers

Address: 0x40020008, Reset: 0x00, Name: GP0PUL

Address: 0x40020048, Reset: 0x00, Name: GP1PUL

Address: 0x40020088, Reset: 0x00, Name: GP2PUL

Table 134. Bit Descriptions for GP0PUL, GP1PUL, and GP2PUL

Bits	Bit Name	Description	Reset	Access
[7:0]	PUL	Pin pull-up enable. 0: disable the pull-up on the corresponding GPIO. 1: enable the pull-up on the corresponding GPIO.	0x00	RW

GPIO Port Input Path Enable Registers

Address: 0x4002000C, Reset: 0x00, Name: GP0IE

Address: 0x4002004C, Reset: 0x00, Name: GP1IE

Address: 0x4002008C, Reset: 0x00, Name: GP2IE

Table 135. Bit Descriptions for GP0IE, GP1IE, and GP2IE

Bits	Bit Name	Description	Reset	Access
[7:0]	IEN	Input path enable. Must be set for external interrupts and to read the pin value. 0: disable the input path on the corresponding GPIO. 1: enable the input path on the corresponding GPIO.	0xFF	RW

GPIO Port Registered Data Input Registers

Address: 0x40020010, Reset: 0xXX, Name: GP0IN

Address: 0x40020050, Reset: 0xXX, Name: GP1IN

Address: 0x40020090, Reset: 0xXX, Name: GP2IN

Table 136. Bit Descriptions for GP0IN, GP1IN, and GP2IN

Bits	Bit Name	Description	Reset	Access
[7:0]	IN	Registered data input. Each bit reflects the state of the GPIO pin.	0xX	R

GPIO Port Data Output Registers

Address: 0x40020014, Reset: 0x0000, Name: GP0OUT

Address: 0x40020054, Reset: 0x0000, Name: GP1OUT

Address: 0x40020094, Reset: 0x0000, Name: GP2OUT

Table 137. Bit Descriptions for GP0OUT, GP1OUT, and GP2OUT

Bits	Bit Name	Description	Reset	Access
[7:0]	OUT	Data out. Do not use the bit-band alias addresses for this register. 0: cleared by user to drive the corresponding GPIO low. 1: set by user code to drive the corresponding GPIO high.	0x0000	RW

GPIO Port Data Out Set Register

Address: 0x40020018, Reset: 0x00, Name: GP0SET

Address: 0x40020058, Reset: 0x00, Name: GP1SET

Address: 0x40020098, Reset: 0x00, Name: GP2SET

Table 138. Bit Descriptions for GP0SET, GP1SET, GP2SET

Bits	Bit Name	Description	Reset	Access
[7:0]	SET	Set the output high. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to drive the corresponding GPIO high.	0x00	W

GPIO Port Data Out Clear Registers

Address: 0x4002001C, Reset: 0x00, Name: GP0CLR

Address: 0x4002005C, Reset: 0x00, Name: GP1CLR

Address: 0x4002009C, Reset: 0x00, Name: GP2CLR

Table 139. Bit Descriptions for GP0CLR, GP1CLR, and GP2CLR

Bits	Bit Name	Description	Reset	Access
[7:0]	CLR	Set the output low. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: each bit is set to drive the corresponding GPIO pin low.	0x00	W

GPIO Port Pin Toggle Registers

Address: 0x40020020, Reset: 0x00, Name: GP0TGL

Address: 0x40020060, Reset: 0x00, Name: GP1TGL

Address: 0x400200A0, Reset: 0x00, Name: GP2TGL

Table 140. Bit Descriptions for GP0TGL, GP1TGL, and GP2TGL

Bits	Bit Name	Description	Reset	Access
[7:0]	TGL	Toggle the output of the port pin. Do not use the bit-band alias addresses for this register. 0: clearing this bit has not effect. 1: set by user code to invert the corresponding GPIO pin.	0x00	W

GPIO Port Open Drain Enable Registers

Address: 0x40020024, Reset: 0x00, Name: GP0ODE

Address: 0x40020064, Reset: 0x00, Name: GP1ODE

Address: 0x400200A4, Reset: 0x00, Name: GP2ODE

Table 141. Bit Descriptions for GP0ODE, GP1ODE, and GP2ODE

Bits	Bit Name	Description	Reset	Access
[7:0]	ODE	Open drain enable. 0: disable the open-drain output mode on corresponding GPIO. 1: enable the open-drain output mode on corresponding GPIO.	0x00	RW

GPIO Port 3 Configuration Register

Address: 0x400200C0, Reset: 0x0000, Name: GP3CON

Table 142. Bit Descriptions for GP3CON

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	RW
[9:8]	CON4	Configuration bits for Port 3.4. See Table 130.	0x0	RW
[7:6]	RESERVED	Reserved.	0x0	RW
[5:4]	CON2	Configuration bits for Port 3.2. See Table 130.	0x0	RW
[3:2]	CON1	Configuration bits for Port 3.1. See Table 130.	0x0	RW
[1:0]	CON0	Configuration bits for Port 3.0. See Table 130.	0x0	RW

GPIO Port 3 Output Enable Register

Address: 0x400200C4, Reset: 0x00, Name: GP3OEN

Table 143. Bit Descriptions for GP3OEN

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	R
4	OEN	Pin output drive enable. 0: disable the output on P3.4. 1: enable the output on P3.4.	0x0	RW
3	RESERVED	Reserved.	0x0	R
2	OEN	Pin output drive enable. 0: disable the output on P3.2. 1: enable the output on P3.2.	0x0	RW
1	OEN	Pin output drive enable. 0: disable the output on P3.1. 1: enable the output on P3.1.	0x0	RW
0	OEN	Pin output drive enable. 0: disable the output on P3.0. 1: enable the output on P3.0.	0x0	RW

GPIO Port 3 Pull-Up Enable Register

Address: 0x400200C8, Reset: 0x00, Name: GP3PUL

Table 144. Bit Descriptions for GP3PUL

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	R
4	PUL	Pin pull-up enable. 0: disable the pull-up on P3.4. 1: enable the pull-up on P3.4.	0x0	RW
3	RESERVED	Reserved.	0x0	R
2	PUL	Pin pull-up enable. 0: disable the pull-up on P3.2. 1: enable the pull-up on P3.2.	0x0	RW
1	PUL	Pin pull-up enable. 0: disable the pull-up on P3.1. 1: enable the pull-up on P3.1.	0x0	RW
0	PUL	Pin pull-up enable. 0: disable the pull-up on P3.0. 1: enable the pull-up on P3.0.	0x0	RW

GPIO Port 3 Input Path Enable Register

Address: 0x400200CC, Reset: 0x00, Name: GP3IE

This register must be set for external interrupts and to read the pin value.

Table 145. Bit Descriptions for GP3IE

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0xE0	R
4	IEN	Input path enable. 0: disable the input path on P3.4. 1: enable the input path on P3.4.	0x1	RW
3	RESERVED	Reserved.	0x1	R
2	IEN	Input path enable. 0: disable the input path on P3.2. 1: enable the input path on P3.2.	0x1	RW
1	IEN	Input path enable. 0: disable the input path on P3.1. 1: enable the input path on P3.1.	0x1	RW
0	IEN	Input path enable. 0: disable the input path on P3.0. 1: enable the input path on P3.0.	0x1	RW

GPIO Port 3 Registered Data Input

Address: 0x400200D0, Reset: 0xXX, Name: GP3IN

Table 146. Bit Descriptions for GP3IN

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0xX	R
4	IN	Registered data input. Reflects the state of P3.4.	0xX	R
3	RESERVED	Reserved.	0xX	R
2	IN	Registered data input. Reflects the state of P3.2.	0xX	R
1	IN	Registered data input. Reflects the state of P3.1.	0xX	R
0	IN	Registered data input. Reflects the state of P3.0.	0xX	R

GPIO Port 3 Data Output Register

Address: 0x400200D4, Reset: 0x0000, Name: GP3OUT

Table 147. Bit Descriptions for GP3OUT

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	R
4	OUT	Data out. Do not use the bit-band alias addresses for this register. 0: cleared by user to drive the corresponding GPIO low. 1: set by user code to drive P3.4 high.	0x0	RW
3	RESERVED	Reserved.	0x0	R
2	OUT	Data out. Do not use the bit-band alias addresses for this register. 0: cleared by user to drive the corresponding GPIO low. 1: set by user code to drive P3.2 high.	0x0	RW
1	OUT	Data out. Do not use the bit-band alias addresses for this register. 0: cleared by user to drive the corresponding GPIO low. 1: set by user code to drive P3.1 high.	0x0	RW
0	OUT	Data out. Do not use the bit-band alias addresses for this register. 0: cleared by user to drive the corresponding GPIO low. 1: set by user code to drive P3.0 high.	0x0	RW

GPIO Port 3 Data Out Set Register

Address: 0x400200D8, Reset: 0x00, Name: GP3SET

Table 148. Bit Descriptions for GP3SET

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	W
4	SET	Set the output high. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to drive P3.4 high.	0x0	W
3	RESERVED	Reserved.	0x0	W
2	SET	Set the output high. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to drive P3.2 high.	0x0	W
1	SET	Set the output high. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to drive P3.1 high.	0x0	W
0	SET	Set the output high. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to drive P3.0 high.	0x0	W

GPIO Port 3 Data Out Clear Register

Address: 0x400200DC, Reset: 0x00, Name: GP3CLR

Table 149. Bit Descriptions for GP3CLR

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	W
4	CLR	Set the output low. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set to drive the P3.4 low.	0x0	W
3	RESERVED	Reserved.	0x0	W
2	CLR	Set the output low. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set to drive the P3.2 low.	0x0	W
1	CLR	Set the output low. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set to drive the P3.1 low.	0x0	W
0	CLR	Set the output low. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set to drive the P3.0 low.	0x0	W

GPIO Port 3 Pin Toggle Register

Address: 0x400200E0, Reset: 0x00, Name: GP3TGL

Table 150. Bit Descriptions for GP3TGL

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	W
4	TGL	Toggle the output of the port pin. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to invert P3.4.	0x0	W
3	RESERVED	Reserved.	0x0	W
2	TGL	Toggle the output of the port pin. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to invert P3.2.	0x0	W
1	TGL	Toggle the output of the port pin. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to invert P3.1.	0x0	W
0	TGL	Toggle the output of the port pin. Do not use the bit-band alias addresses for this register. 0: clearing this bit has no effect. 1: set by user code to invert P3.0.	0x0	W

GPIO Port 3 Open Drain Enable Register

Address: 0x400200E4, Reset: 0x00, Name: GP3ODE

Table 151. Bit Descriptions for GP3ODE

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x00	RW
4	ODE	Open-drain enable. 0: disable the open-drain output mode on P3.4. 1: enable the open-drain output mode on P3.4.	0x0	RW
3	RESERVED	Reserved.	0x0	RW
2	ODE	Open-drain enable. 0: disable the open-drain output mode on P3.2. 1: enable the open-drain output mode on P3.2.	0x0	RW
1	ODE	Open-drain enable. 0: disable the open-drain output mode on P3.1. 1: enable the open-drain output mode on P3.1.	0x0	RW
0	ODE	Open-drain enable. 0: disable the open-drain output mode on P3.0. 1: enable the open-drain output mode on P3.0.	0x0	RW

I²C SERIAL INTERFACE

I²C FEATURES

The I²C interface has the following features:

- Master or slave mode with 2-byte transmit and receive FIFOs
- Supports
    - 7-bit and 10-bit addressing modes
    - Four 7-bit device addresses or one 10-bit address and two 7-bit addresses in the slave
    - Repeated starts in master and slave modes
    - Clock stretching supported for the slave and master
    - Master arbitration
    - Continuous read mode for the master or up to 512 bytes fixed read
    - Internal and external loopback
- Support for DMA in master and slave modes
- Software control on the slave of no acknowledge (NACK) signal

I²C OVERVIEW

The I²C data transfer uses a serial clock pin (SCL) and a serial data pin (SDA). The pins are configured in a wired-AND'ed format that allows arbitration in a multiple master system.

The transfer sequence of an I²C system consists of a master device initiating a transfer by generating a start condition while the bus is idle. The master transmits the slave device address and the direction of the data transfer during the initial address transfer. If the master does not lose arbitration and the slave acknowledges the initial address transfer, the data transfer is initiated. This continues until the master issues a stop condition and the bus becomes idle. Figure 20 shows a typical I²C transfer.

A master device can be configured to generate the serial clock. The user programs the frequency in the serial clock divisor register, I2CxDIV (where x is 0 for I²C0 and 1 for I²C1). The master channel can be set to operate in fast mode (400 kHz) or in standard mode (100 kHz).

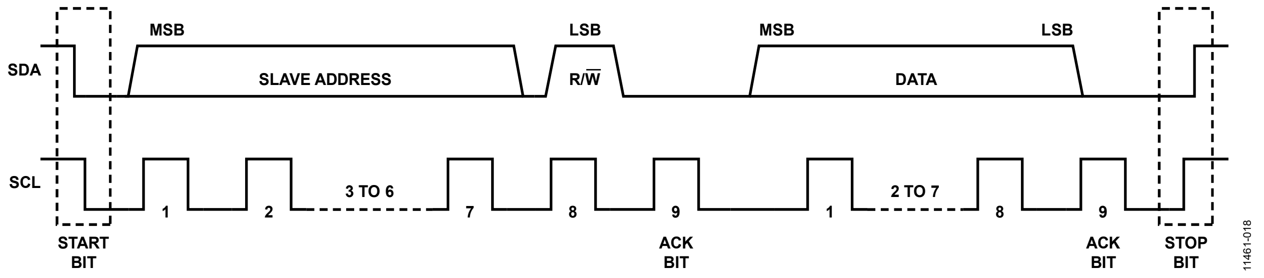


Figure 20. Typical I²C Transfer Sequence

The user programs the I²C bus peripheral address in the I²C bus system. This ID can be modified any time a transfer is not in progress. The user can set up to four slave addresses that are recognized by the peripheral. The peripheral is implemented with a 2-byte FIFO for each transmit and receive shift register. The IRQ and status bits in the control registers are available to signal to the processor core when the FIFOs must be serviced.

I²C OPERATION

I²C Startup

The following steps are required to set the I²C peripheral running:

1. Configure I²C clock in CLKCON1[10:8], CLKCON5[4] for I²C1, and CLKCON5[3] for I²C0.
2. Configure digital pins for I²C operation via the GP0CON register (P0.4/P0.5, P0.6/P0.7).
3. Configure I²C registers as required for slave or master operation.
4. Enable the I²C slave or master interrupt source as required.

Note that the user must disable the internal pull-up resistors on the I²C pins via the GP0PUL register when using I²C.

Addressing Modes

7-Bit Addressing

The I2CxID0, I2CxID1, I2CxID2, and I2CxID3 registers contain the slave device IDs. The device compares the four I2CxIDx registers to the address byte. To be correctly addressed, the seven MSBs of either ID register must be identical to that of the seven MSBs of the first received address byte. The LSB of the ID registers (the transfer direction bit) is ignored in the process of address recognition.

The master addresses a device using the I2CxADR0 register.

10-Bit Addressing

This feature is enabled by setting I2CxSCON[1] for master and slave mode.

The 10-bit address of the slave is stored in I2CxID0 and I2CxID1, where I2CxID0 contains the first byte of the address, and the R/$\overline{\text{W}}$ bit and the upper five bits must be programmed to 11110, as shown in Figure 21. I2CxID1 contains the remaining eight bits of the 10-bit address. I2CxID2 and I2CxID3 can still be programmed with 7-bit addresses.

The master communicates to a 10-bit address slave using the I2CxADR0 and I2CxADR1 registers. The format is shown in Figure 21.

I2CADR0 AND I2CID0								I2CADR1 AND I2CID1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	1	1	0	2 MSBs		R/$\overline{\text{W}}$	8 LSBs							

11461-019

Figure 21. 10-Bit Address Format

Repeated Start Condition

A repeated start condition occurs when a second start condition is sent to a slave without a stop condition being sent in between. This allows the master to reverse the direction of the transfer by changing the R/$\overline{\text{W}}$ bit without having to give up control of the bus.

An example of a transfer sequence is shown in Figure 22. This is generally used where the first data sent to the device sets up the register address to be read from.

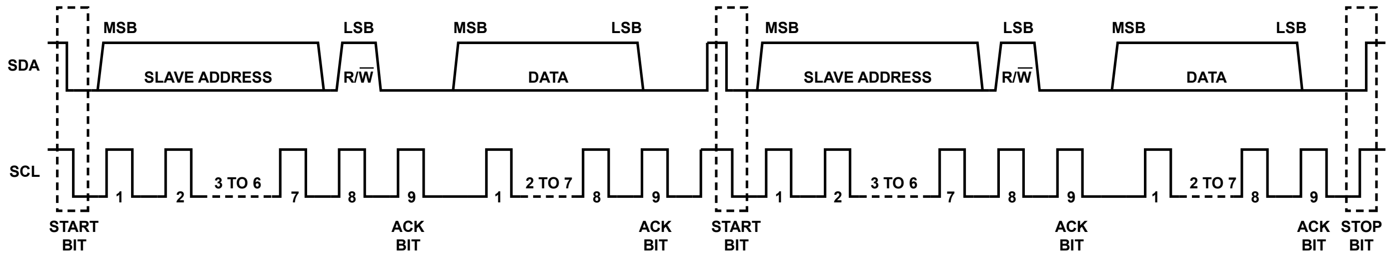


Figure 22. I^{2}C Repeated Start Sequence

On the slave side, an interrupt is generated (if enabled in the I2CxSCON register) when a repeated start and a slave address are received. This can be differentiated from receiving a start and slave address using the START and REPSTART status bits in the I2CxSSTA MMR.

On the master side, the master generates a repeated start if the I2CxADR0 register is written while the master is still busy with a transaction. After the state machine starts to transmit the device address, it is safe to write to the I2CxADR0 register.

For example, if a transaction involving a write, repeated start, and then read/write is required, write to the I2CxADR0 register either after the state machine starts to transmit the device address or after the first MTXREQ interrupt is received. When the transmit FIFO empties, a repeated start is generated.

Similarly, if a transaction involving a read, repeated start, and then read/write is required, write to the first master address byte register, I2CxADR0, either after the state machine starts to transmit the device address or after the first MRXREQ interrupt is received. When the requested receive count is reached, a repeated start is generated.

I²C Clock Control

The I²C peripherals are clocked by a gated 20 MHz system clock (PCLK). The CLKCON5[3] bit must be cleared to enable the clock to the I²C0 block. Similarly, the CLKCON5[4] bit must be cleared to enable the clock to the I²C1 block. CLKCON1[10:8] allow the I²C block to be clocked with a slower clock by allowing the 20 MHz clock to be divided, which helps to reduce power.

The I²C master in the system generates the serial clock for a transfer. The master channel can be configured to operate in fast mode (400 kHz) or in standard mode (100 kHz).

The bit rate is defined in the I2CxDIV MMR as follows:

$$f_{SCL} = f_{I2CCLK}/(LOW + HIGH + 3)$$

where:
$f_{I2CCLK} = f_{PCLK}/I2CCD$.
f_{PCLK} is the system clock, 20 MHz.
I2CCD is the clock divide value and is set by the CLKCON1[10:8].
HIGH is the high period of the clock, I2CxDIV[15:8] = (REQD_HIGH_TIME/UCLK_PERIOD) − 2.
LOW is the low period of the clock, I2CxDIV[7:0] = (REQD_LOW_TIME/UCLK_PERIOD) − 1.

For 100 kHz SCL operation, with a low time of 5 µs and a high time of 5 µs, and a UCLK frequency of 20 MHz,

$$HIGH = (5\ \mu s\ /(1/20{,}000{,}000)) - 2 = 98 = 0x62$$

$$LOW = (5\ \mu s\ /(1/20{,}000{,}000)) - 1 = 99 = 0x63$$

$$f_{SCL} = 20{,}000{,}000/(98 + 99 + 3) = 100\ kHz$$

I²C OPERATING MODES

Master Transfer Initiation

If the master enable bit (I2CxMCON[0], MASEN) is set, a master transfer sequence is initiated by writing a value to the I2CxADRx register. If there is valid data in the I2CxMTX register, it is the first byte transferred in the sequence after the address byte during a write sequence.

Slave Transfer Initiation

If the slave enable bit (I2CxSCON[0], SLVEN) is set, a slave transfer sequence is monitored for the device address in Register I2CxID0, Register I2CxID1, Register I2CxID2, or Register I2CxID3. If the device address is recognized, the device participates in the slave transfer sequence.

Note that a slave operation always starts with the assertion of one of three interrupt sources: read request (MRXREQ, SRXREQ), write request (MTXREQ, STXREQ), or general call (GCINT) interrupt. The software must always look for a stop interrupt to ensure that the transaction has completed correctly and to deassert the stop interrupt status bit.

Rx/Tx Data FIFOs

The transmit data path consists of a master and slave Tx FIFO, I2CxMTX and I2CxSTX (each two bytes deep), and a transmit shifter. The transmit status bits in I2CxMSTA[1:0] and I2CxSSTA[0] denote whether there is valid data in the Tx FIFO. Data from the Tx FIFO is loaded into the Tx shifter when a serial byte begins transmission. If the Tx FIFO is not full during an active transfer sequence, the transmit request bit (I2CxMSTA[2] or I2CxSSTA[2]) asserts.

Figure 23 shows the effect of not having the slave Tx FIFO full at the start of a read request from a master. An extra transmit interrupt may be generated after the read bit. This extra transmit interrupt occurs if the Tx FIFO is not full.

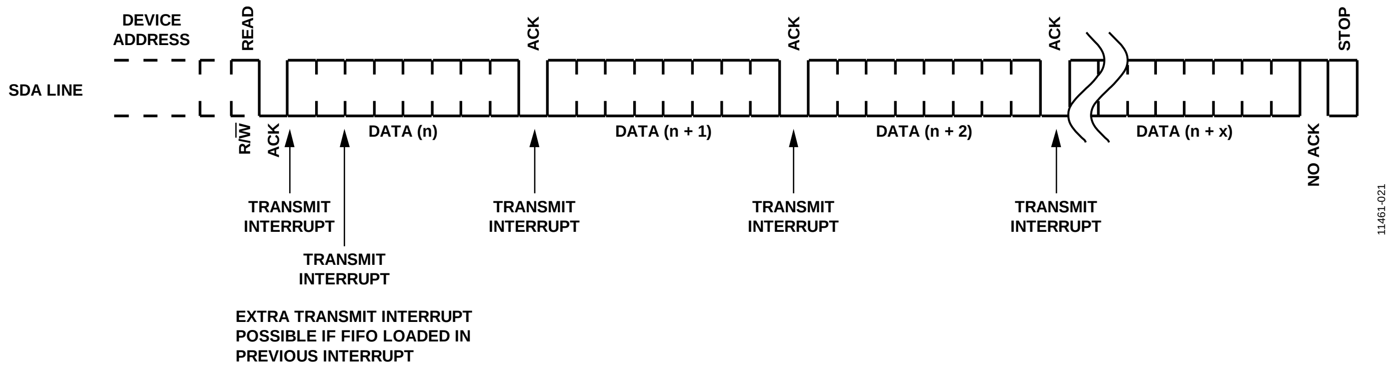


Figure 23. I²C Slave Tx Interrupt Details

In the slave, if there is no valid data to transmit when the Tx shifter is loaded, the transmit underflow status bit asserts (I2CxMSTA[12], ISCxSSTA[1]). In slave mode, the Tx FIFO must be loaded with a byte before the falling edge of SCL before the acknowledge/no acknowledge is asserted.

If the Tx FIFO is empty on the falling edge of SCL for a R/W bit, the slave returns a no acknowledge because the slave in this case controls the acknowledge/no acknowledge.

If the first byte is transmitted correctly in a slave Tx sequence but the Tx FIFO is empty for any subsequent bytes in the same transfer, the slave returns the previous transmitted byte. This operation is due to the master having control of the acknowledge/no acknowledge during a slave transfer sequence.

The master generates a stop condition if there is no data in the transmit FIFO and the master is writing data.

The receive data path consists of a master and slave Rx FIFO (I2CxMRX and I2CxSRX), each two bytes deep. The receive request interrupt bit (I2CxMSTA[3] or I2CxSSTA[3]) indicates whether there is valid data in the Rx FIFO. Data is loaded into the Rx FIFO after each byte is received. If valid data in the Rx FIFO is overwritten by the Rx shifter, the receive overflow status bit is asserted (I2CxMSTA[9] or I2CxSSTA[4]).

Automatic Clock Stretching

The ADuCM310 supports automatic clock stretching in both master and slave modes.

It is recommended that automatic clock stretching be enabled, especially in slave mode.

A timeout feature is added to ensure that the I^2C block never erroneously holds the SCL pin low indefinitely. A separate status pin for master and slave mode indicates if stretch timeout occurred.

The I2CxASSCL register controls automatic clock stretching. If automatic clock stretching is enabled, the I^2C hardware holds the SCL pin low after the falling edge of SCL before an acknowledge/no acknowledge during the following conditions:

- Tx FIFO is empty when a valid read request is active for the master or slave.
  - If at the end of the timeout period, the Tx FIFO is still empty, the following occurs:
    - If the Tx FIFO is empty on the falling edge of SCL for a R/W bit, the slave returns a no acknowledge after the timeout period.
    - If the first byte is transmitted correctly in a slave Tx sequence but the Tx FIFO is empty for any subsequent bytes in the same transfer with clock stretch enabled, the slave returns the previous transmitted byte at the end of the timeout period.
- Rx FIFO is full when another byte is about to be received. If the Rx FIFO has still not been read at the end of the timeout period, a no acknowledge is returned and the master ends the sequence with a stop condition.

It is not recommended to use the I2CxSCON[6] clock stretching method when using automatic clock stretching.

Master No Acknowledge

When receiving data, the master responds with a no acknowledge if its FIFO is full and an attempt is made to write another byte to the FIFO. This last byte received is not written to the FIFO and is lost.

No Acknowledge from the Slave

If the slave does not want to acknowledge a read access, then simply not writing data into the slave transmit FIFO results in a no acknowledge.

If the slave does not want to acknowledge a master write, assert the no acknowledge bit (NACK) in the slave control register, I2CxSCON[7].

Normally, the slave acknowledges all bytes that are written into the receive FIFO. If the receive FIFO fills up, the slave cannot write further bytes to it, and it does not acknowledge the byte that was not written to the FIFO. The master must then stop the transaction.

The slave does not acknowledge a matching device address if the read/write bit is set and the transmit FIFO is empty. Therefore, there is very little time for the microcontroller to respond to a slave transmit request and the assertion of the acknowledge. It is recommended that EARLYTXR, I2CxSCON[5], be asserted for this reason.

General Call

An I^2C general call is for addressing every device on the I^2C bus. A general call address is 0x00 or 0x01. The first byte, the address byte, is followed by a command byte.

If the address byte is 0x00, Byte 2 (the command byte) can be one of the following:

- 0x6: the I^2C interface (master and slave) is reset. The general call interrupt status asserts, and the general call ID bits, GCID (I2CxSSTA[9:8]), are 0x1. User code must take corrective action to reset the entire system or simply to reenable the I^2C interface.
- 0x4: the general call interrupt status bit is asserted, and the general call ID bits (GCID) are 0x2.

If the address byte is 0x01, a hardware general call is issued. Byte 2 in this case is the hardware master address.

The general call interrupt status bit is set on any general call after the second byte is received, and user code must take corrective action to reprogram the device address.

If GCEN is asserted, the slave always acknowledges the first byte of a general call. It acknowledges the second byte of a general call if the second byte is 0x04 or 0x06, or if the second byte is a hardware general call, and HGCEN (I2CxSCON[3]) is asserted.

The I2CxALT register contains the alternate device ID for a hardware general call sequence. If the hardware general call enable bit, HGCEN, GCEN, and SLVEN are all set, the device recognizes a hardware general call. When a general call sequence is issued and the second byte of the sequence is identical to ALT, the hardware call sequence is recognized for the device.

I^{2}C Reset Mode

The slave state machine is reset when SLVEN is written to 0.

The master state machine is reset when MASEN is written to 0.

I^{2}C Test Modes

The device can be placed in an internal loopback mode by setting the LOOPBACK bit (I2CxMCON[2]). There are four FIFOs (master Tx and Rx, and slave Tx and Rx); therefore, in effect, the I^{2}C peripheral can be set up to talk to itself. External loopback can be performed if the master is set up to address the slave address.

I^{2}C Low Power Mode

If the master and slave are both disabled (MASEN = SLVEN = 0), the I^{2}C section is off. To fully power down the I^{2}C block, disable the clock to the I^{2}C section of the chip by setting CLKCON5[4:3] = 0x3.

DMA Requests

Four DMA channels are required to service the I^{2}C master and slave. DMA enable bits are provided in the slave control register and in the master control register.

REGISTER SUMMARY: I^{2}C0

Table 152. I^{2}C0 Register Summary

Address	Name	Description	Reset	Access
0x40003000	I2CMCON	Master control register	0x0000	RW
0x40003004	I2CMSTA	Master status register	0x6000	R
0x40003008	I2CMRX	Master receive data register	0x0000	R
0x4000300C	I2CMTX	Master transmit data register	0x0000	RW
0x40003010	I2CMRXCNT	Master receive data count register	0x0000	RW
0x40003014	I2CMCRXCNT	Master current receive data count register	0x0000	R
0x40003018	I2CADR0	First master address byte register	0x0000	RW
0x4000301C	I2CADR1	Second master address byte register	0x0000	RW
0x40003024	I2CDIV	Serial clock period divisor register	0x1F1F	RW
0x40003028	I2CSCON	Slave control register	0x0000	RW
0x4000302C	I2CSSTA	Slave I^{2}C status/error/IRQ register	0x0001	R
0x40003030	I2CSRX	Slave receive register	0x0000	R
0x40003034	I2CSTX	Slave transmit register	0x0000	RW
0x40003038	I2CALT	Hardware general call ID register	0x0000	RW
0x4000303C	I2CID0	First slave address device ID register	0x0000	RW
0x40003040	I2CID1	Second slave address device ID register	0x0000	RW
0x40003044	I2CID2	Third slave address device ID register	0x0000	RW
0x40003048	I2CID3	Fourth slave address device ID register	0x0000	RW
0x4000304C	I2CFSTA	Master and slave FIFO status register	0x0000	RW
0x40003050	I2C0SHCON	Master and slave shared control register	0x0000	W
0x40003058	I2CASSCL	Automatic stretch control register for master and slave mode	0x0000	RW

REGISTER DETAILS: I^{2}C0

Master Control Register

Address: 0x40003000, Reset: 0x0000, Name: I2CMCON

Table 153. Bit Descriptions for I2CMCON

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
11	MTXDMA	Enable master Tx DMA request. 0: disable DMA mode. 1: enable I^{2}C master DMA Tx requests.	0x0	W
10	MRXDMA	Enable master Rx DMA request. 0: disable DMA mode. 1: enable I^{2}C master DMA Rx requests.	0x0	W
9	RESERVED	Reserved.	0x0	RW
8	IENCMP	Transaction completed (or stop detected) interrupt enable. 0: an interrupt is not generated when a stop is detected. 1: an interrupt is generated when a stop is detected.	0x0	RW
7	IENACK	Acknowledge not received interrupt enable. 0: disable acknowledge not received interrupt. 1: enable acknowledge not received interrupt.	0x0	RW
6	IENALOST	Arbitration lost interrupt enable. 0: disable arbitration lost interrupt. 1: enable arbitration lost interrupt.	0x0	RW
5	IENMTX	Transmit request interrupt enable. 0: disable transmit request interrupt. 1: enable transmit request interrupt.	0x0	RW
4	IENMRX	Receive request interrupt enable. 0: disable receive request interrupt. 1: enable receive request interrupt.	0x0	RW
3	STRETCH	Stretch SCL enable. 0: disable clock stretching. 1: setting this bit tells the device if SCL is 0, hold it at 0; or if SCL is 1, hold SCL at 0 when it next goes to 0.	0x0	RW
2	LOOPBACK	Internal loopback enable. Note that is also possible for the master to loop back a transfer to the slave as long as the device address corresponds, that is, external loopback. 0: SCL and SDA out of the device are not muxed onto their corresponding inputs. 1: SCL and SDA out of the device are muxed onto their corresponding inputs.	0x0	RW
1	COMPETE	Start backoff disable. Setting this bit enables the device to compete for ownership even if another device is currently driving a start condition.	0x0	RW
0	MASEN	Master enable. Disable the master when not in use, which gates the clock to the master and saves power. Do not clear this bit until a transaction has completed; see the TCOMP bit in the master status register. 0: master is disabled. 1: master is enabled.	0x0	RW

Master Status Register

Address: 0x40003004, Reset: 0x6000, Name: I2CMSTA

Table 154. Bit Descriptions for I2CMSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	SCL_FILTERED	State of SCL line. This bit is the output of the glitch filter on SCL. SCL is always pulled high when undriven.	0x1	R
13	SDA_FILTERED	State of SDA line. This bit is the output of the glitch filter on SDA. SDA is always pulled high when undriven.	0x1	R
12	MTXUFLOW	Master transmit underflow. Asserts when the I^{2}C master ends the transaction due to Tx FIFO empty condition. This bit is asserted only when the IENMTX bit is set.	0x0	RC
11	MSTOP	Stop driven by this I^{2}C Master. Asserts when this I^{2}C master drives a stop condition on the I^{2}C bus. This bit, when asserted, can indicate a transaction completion, Tx underflow, Rx overflow, or a no acknowledge by the slave. This is different from the TCOMP because this bit is not asserted when the stop condition occurs due to any other I^{2}C master. No interrupt is generated for the assertion of this bit. However, if IENCMP is 1, every stop condition generates an interrupt, and this bit can be read. When this bit is read, it clears status.	0x0	RC
10	LINEBUSY	Line is busy. Asserts when a start is detected on the I^{2}C bus. Deasserts when a stop is detected on the I^{2}C bus.	0x0	R
9	MRXOF	Master receive FIFO overflow. Asserts when a byte is written to the receive FIFO when the FIFO is already full. When the bit is read, it clears status.	0x0	RC
8	TCOMP	Transaction complete or stop detected. This bit asserts when a stop condition is detected on the I^{2}C bus. If IENCMP is 1, an interrupt is generated when this bit asserts. This bit only asserts if the master is enabled (MASEN = 1). Use this bit to determine when it is safe to disable the master. It can also wait for another master transaction to complete on the I^{2}C bus when this master loses arbitration. When this bit is read, it clears status. This bit can drive an interrupt.	0x0	RC
7	NACKDATA	Acknowledge not received in response to data write. This bit asserts when an acknowledge is not received in response to a data write transfer. If IENACK is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt. This bit is cleared on a read of the I2C0MSTA register.	0x0	RC
6	MBUSY	Master busy. This bit indicates that the master state machine is servicing a transaction. It is clear if the state machine is idle or if another device has control of the I^{2}C bus.	0x0	R
5	ALOST	Arbitration lost. This bit asserts if the master loses arbitration. If IENALOST is 1, an interrupt is generated when this bit asserts. This bit is cleared on a read of the I2C0MSTA register. This bit can drive an interrupt.	0x0	RC
4	NACKADDR	Acknowledge not received in response to an address. This bit asserts if an acknowledge is not received in response to an address. If IENACK is 1, an interrupt is generated when this bit asserts. This bit is cleared on a read of the I2CMSTA register. This bit can drive an interrupt.	0x0	RC
3	MRXREQ	Master receive request. This bit asserts when there is data in the receive FIFO. If IENMRX is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt.	0x0	R
2	MTXREQ	Master transmit request. This bit asserts when the direction bit is 0 and the transmit FIFO is either empty or not full. If IENMTX is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt.	0x0	R
[1:0]	MTXFSTA	Master transmit FIFO status. These two bits show the master transmit FIFO status and can be decoded as follows: 00 = FIFO empty. 10 = 1 byte in FIFO. 11 = FIFO full.	0x0	R

Master Receive Data Register

Address: 0x40003008, Reset: 0x0000, Name: I2CMRX

Table 155. Bit Descriptions for I2CMRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ICMRX	Master receive register. This register allows access to the receive data FIFO. The FIFO can hold two bytes.	0x0	R

Master Transmit Data Register

Address: 0x4000300C, Reset: 0x0000, Name: I2CMTX

Table 156. Bit Descriptions for I2CMTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	I2CMTX	Master transmit register. For test and debug purposes, when read, this register returns the byte that is currently being transmitted by the master. That is a byte written to the transmit register can be read back some time later when that byte is being transmitted on the line. This register allows access to the transmit data FIFO. The FIFO can hold two bytes.	0x0	RW

Master Receive Data Count Register

Address: 0x40003010, Reset: 0x0000, Name: I2CMRXCNT

Table 157. Bit Descriptions for I2CMRXCNT

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	EXTEND	Extended read. Use this bit if more than 256 bytes are required on a read. For example, to receive 412 bytes, write 0x100 (EXTEND = 1) to the I2CMRXCNT register. Wait for the first byte to be received, then check the I2CMCRXCNT register for every byte received thereafter. When COUNT returns to 0, 256 bytes have been received. Then write 0x09C to the I2CMRXCNT register.	0x0	RW
[7:0]	COUNT	Receive count. Program the number of bytes required minus one to this register. If just 1 byte is required, write 0 to this register. If more than 256 bytes are required, use EXTEND.	0x0	RW

Master Current Receive Data Count Register

Address: 0x40003014, Reset: 0x0000, Name: I2CMCRXCNT

Table 158. Bit Descriptions for I2CMCRXCNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	COUNT	Current receive count. This register gives the total number of bytes received so far. If 256 bytes are requested, this register reads 0 when the transaction completes.	0x0	R

First Master Address Byte Register

Address: 0x40003018, Reset: 0x0000, Name: I2CADR0

Table 159. Bit Descriptions for I2CADR0

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ADR0	Address Byte 0. If a 7-bit address is required, Bit 7 to Bit 1 of ADR0 are programmed with the address, and Bit 0 of ADR0 is programmed with the direction (0 = write, 1 = read). If a 10-bit address is required, Bit 7 to Bit 3 of ADR0 are programmed with 11110, Bit 2 to Bit 1 of ADR0 are programmed with the 2 MSBs of the address, and Bit 0 of ADR0 is programmed to 0.	0x0	RW

Second Master Address Byte Register

Address: 0x4000301C, Reset: 0x0000, Name: I2CADR1

Table 160. Bit Descriptions for I2CADR1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ADR1	Address Byte 1. This register is only required when addressing a slave with a 10-bit address. Bit 7 to Bit 0 of ADR1 are programmed with the lower 8 bits of the address.	0x0	RW

Serial Clock Period Divisor Register

Address: 0x40003024, Reset: 0x1F1F, Name: I2CDIV

Table 161. Bit Descriptions for I2CDIV

Bits	Bit Name	Description	Reset	Access
[15:8]	HIGH	Serial clock high time. This register controls the clock high time. The timer is driven by the core clock (UCLK). Use the following equation to derive the required high time: *HIGH* = (*REQD_HIGH_TIME*/*UCLK_PERIOD*) − 2 For example, to generate a 400 kHz SCL with a low time of 1300 ns and a high time of 1200 ns, with a core clock frequency of 50 MHz: *LOWTIME* = 1300 ns/20 ns − 1 = 0x40 (64 decimal) *HIGH* = 1200 ns/20 ns − 2 = 0x3A (58 decimal) This register is reset to 0x1F, which gives an SCL high time of 33 UCLK ticks. $t_{HD:STA}$ is also determined by the high time. $t_{HD:STA}$ = (*HIGH* − 1) × *UCLK_PERIOD* Because $t_{HD:STA}$ must be 600 ns, with UCLK = 50 MHz, the minimum value for HIGH is 31. This gives an SCL high time of 660 ns.	0x1F	RW
[7:0]	LOW	Serial clock low time. This register controls the clock low time. The timer is driven by the core clock (UCLK). Use the following equation to derive the required low time. *LOW* = (*REQD_LOW_TIME*/*UCLK_PERIOD*) − 1 This register is reset to 0x1F, which gives an SCL low time of 32 UCLK ticks.	0x1F	RW

Slave Control Register

Address: 0x40003028, Reset: 0x0000, Name: I2CSCON

Table 162. Bit Descriptions for I2CSCON

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	STXDMA	Enable slave Tx DMA request. Set to 1 by user code to enable I^{2}C slave DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	RW
13	SRXDMA	Enable slave Rx DMA request. Set to 1 by user code to enable I^{2}C slave DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	RW
12	IENREPST	Repeated start interrupt enable. If 1, an interrupt is generated when the REPSTART status bit asserts. If 0, an interrupt is not generated when the REPSTART status bit asserts.	0x0	RW
11	SXMITDEC	Decrement slave Tx FIFO status when a byte has been transmitted. If set to 1, the transmit FIFO status is decremented when a byte has been transmitted. If set to 0, the transmit FIFO status is decremented when the byte is unloaded from the FIFO into a shadow register at the start of byte transmission.	0x0	RW
10	IENSTX	Slave transmit request interrupt enable.	0x0	RW
9	IENSRX	Slave receive request interrupt enable.	0x0	RW
8	IENSTOP	Stop condition detected interrupt enable.	0x0	RW
7	NACK	No acknowledge next communication. If this bit is set, the next communication is not acknowledged. This can be used, for example, if during a 24xx style access, an attempt was made to write to a read only or nonexistent location in system memory. That is the indirect address in a 24xx style write pointed to an unwritable memory location.	0x0	RW
6	STRETCHSCL	Stretch SCL enable. Setting this bit tells the device, if SCL is 0, hold it at 0; or, if SCL is 1, hold it at 0 when it next goes to 0.	0x0	RW
5	EARLYTXR	Early transmit request mode. Setting this bit enables a transmit request just after the positive edge of the direction bit SCL clock pulse.	0x0	RW
4	GCSBCLR	General call status bit clear. The general call status and general call ID bits are cleared when a 1 is written to this bit. The general call status and general call ID bits are not reset by anything other than a write to this bit or a full reset.	0x0	W
3	HGCEN	Hardware general call enable. When this bit and the general call enable bit are set, the device after receiving a general call, Address 00h, and a data byte, checks the contents of the ALT against the receive shift register. If they match, the device has received a hardware general call. When a device requires urgent attention from a master device without knowing which master it requires to turn to, it can use this call. This is a call "to whom it may concern." The device that requires attention embeds its own address into the message. The LSB of the ALT register must always be written to a 1, as per I^{2}C January 2000 specification.	0x0	RW
2	GCEN	General call enable. This bit enables the I^{2}C slave to acknowledge an I^{2}C general call, Address 0x00 (write).	0x0	RW
1	ADR10EN	Enabled 10-bit addressing. If this bit is clear, the slave can support four slave addresses, programmed in Register I2CID0 to Register I2CID3. When this bit is set, 10-bit addressing is enabled. One 10-bit address is supported by the slave and is stored in I2CID0 and I2CID1, where I2CID0 contains the first byte of the address and the upper five bits must be programmed to 11110. I2CID3 and I2CID4 can be programmed with 7-bit addresses at the same time.	0x0	RW
0	SLVEN	Slave enable. When 1, the slave is enabled. When 0, all slave state machine flops are held in reset, and the slave is disabled. Note that APB writable register bits are not reset.	0x0	RW

Slave I^{2}C Status/Error/IRQ register

Address: 0x4000302C, Reset: 0x0001, Name: I2CSSTA

Table 163. Bit Descriptions for I2CSSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	START	Start and matching address. This bit is asserted if a start is detected on SCL/SDA and the device address matched; if a general call (address = 0000_0000) code is received and general call is enabled; if a high speed (address = 0000_1XXX) code is received; or if a start byte (0000_0001) is received. It is cleared on receipt of either a stop or start condition.	0x0	R
13	REPSTART	Repeated start and matching address. This bit is asserted if a start is already asserted and then a repeated start is detected. It is cleared when read or on receipt of a stop condition. This bit can drive an interrupt.	0x0	RC
[12:11]	IDMAT	Device ID matched. 00: received address matched ID Register 0. 01: received address matched ID Register 1. 10: received address matched ID Register 2. 11: received address matched ID Register 3.	0x0	R
10	STOP	Stop after start and matching address. This bit is set by hardware if the slave device received a stop condition after a previous start condition and a matching address. It is cleared by a read of the status register. If STOPINTEN in the slave control register is asserted, the slave interrupt request asserts when this bit is set. This bit can drive an interrupt.	0x0	RC
[9:8]	GCID	General ID. GCID is cleared when the GCSBCLR is written to 1. These status bits are not cleared by a general call reset. 00: no general call. 01: general call reset and program address. 10: general call program address. 11: general call matching alternative ID.	0x0	R
7	GCINT	General call interrupt. This bit always drives an interrupt. The bit is asserted if the slave device receives a general call of any type. To clear, write 1 to the GCSBCLR in the slave control register. If it was a general call reset, all registers are at their default values. If it was a hardware general call, the Rx FIFO holds the second byte of the general call, and this can be compared with the ALT register.	0x0	R
6	SBUSY	Slave busy. Set by hardware if the slave device receives an I^{2}C start condition. Cleared by hardware when the address does not match an ID register, if the slave device receives a I^{2}C stop condition, or if a repeated start address does not match.	0x0	R
5	NACK	Acknowledge not generated by the slave. When asserted, it indicates that the slave responded to its device address with a no acknowledge. It is asserted if there was no data to transmit and the sequence was a slave read, or if the no acknowledge bit was set in the slave control register and the device was addressed. This bit is cleared on a read of the I2CSSTA register.	0x0	RC
4	SRXOF	Slave receive FIFO overflow. Asserts when a byte is written to the slave receive FIFO when the FIFO is already full.	0x0	RC
3	SRXREQ	Slave receive request. SRXREQ asserts whenever the slave receive FIFO is not empty. Read or flush the slave receive FIFO to clear this bit. This bit asserts on the falling edge of the SCL clock pulse that clocks in the last data bit of a byte. This bit can drive an interrupt.	0x0	RC
2	STXREQ	Slave transmit request. If EARLYTXR = 0, STXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted on the negative edge of the SCL pulse that clocks in the direction bit (if the device address matched also). If EARLYTXR = 1, STXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted after the positive edge of the SCL pulse that clocks in the direction bit (if the device address also matched). This bit is cleared on a read of the I2CSSTA register.	0x0	RC

Bits	Bit Name	Description	Reset	Access
1	STXUR	Slave transmit FIFO underflow. This bit is set if a master requests data from the device, and the Tx FIFO is empty for the rising edge of SCL.	0x0	RC
0	STXFSEREQ	Slave Tx FIFO status or early request. If EARLYTXR = 0, this bit is asserted whenever the slave Tx FIFO is empty. If EARLYTXR = 1, TXFSEREQ is set when the direction bit for a transfer is received high. It asserts on the positive edge of the SCL clock pulse that clocks in the direction bit (if the device address matched also). It only asserts once for a transfer. It is cleared when read if EARLYTXR is asserted.	0x1	RW

Slave Receive Register

Address: 0x40003030, Reset: 0x0000, Name: I2CSRX

Table 164. Bit Descriptions for I2CSRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved	0x0	R
[7:0]	I2CSRX	Slave receive register	0x0	R

Slave Transmit Register

Address: 0x40003034, Reset: 0x0000, Name: I2CSTX

Table 165. Bit Descriptions for I2CSTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved	0x0	R
[7:0]	I2CSTX	Slave transmit register	0x0	RW

Hardware General Call ID Register

Address: 0x40003038, Reset: 0x0000, Name: I2CALT

Table 166. Bit Descriptions for I2CALT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ALT	Slave alt. This register is used in conjunction with I2CSCON[3] to match a master generating a hardware general call. It is used when a master device cannot be programmed with a slave address and instead the slave must recognize the master address.	0x0	RW

First Slave Address Device ID Register

Address: 0x4000303C, Reset: 0x0000, Name: I2CID0

Table 167. Bit Descriptions for I2CID0

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID0	Slave Device ID 0. I2CID0[7:1] is programmed with the device ID. I2CID0[0] is don't care. See the ADR10EN bit in the slave control register for how this register is programmed with a 10-bit address.	0x0	RW

Second Slave Address Device ID Register

Address: 0x40003040, Reset: 0x0000, Name: I2CID1

Table 168. Bit Descriptions for I2CID1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID1	Slave Device ID 1. I2CID1[7:1] is programmed with the device ID. I2CID1[0] is don't care. See the ADR10EN bit in the slave control register for how this register is programmed with a 10-bit address.	0x0	RW

Third Slave Address Device ID Register

Address: 0x40003044, Reset: 0x0000, Name: I2CID2

Table 169. Bit Descriptions for I2CID2

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID2	Slave Device ID 2. I2CID2[7:1] is programmed with the device ID. I2CID2[0] is don't care. See the ADR10EN bit in the slave control register for how this register is programmed with a 10-bit address.	0x0	RW

Fourth Slave Address Device ID Register

Address: 0x40003048, Reset: 0x0000, Name: I2CID3

Table 170. Bit Descriptions for I2CID3

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID3	Slave Device ID 3. I2CID3[7:1] is programmed with the device ID. I2CID3[0] is don't care. See the ADR10EN bit in the slave control register for how this register is programmed with a 10-bit address.	0x0	RW

Master and Slave FIFO Status Register

Address: 0x4000304C, Reset: 0x0000, Name: I2CFSTA

Table 171. Bit Descriptions for I2CFSTA

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	RW
9	MFLUSH	Flush the master transmit FIFO. 0: clearing to 0 has no effect. 1: set to 1 to flush the master transmit FIFO. The master transmit FIFO must be flushed if arbitration is lost or if a slave responds with a no acknowledge.	0x0	W
8	SFLUSH	Flush the slave transmit FIFO. 0: clearing to 0 has no effect. 1: set to 1 to flush the slave transmit FIFO.	0x0	W
[7:6]	MRXFSTA	Master receive FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R
[5:4]	MTXFSTA	Master transmit FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R
[3:2]	SRXFSTA	Slave receive FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R
[1:0]	STXFSTA	Slave transmit FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R

Master and Slave Shared Control Register

Address: 0x40003050, Reset: 0x0000, Name: I2C0SHCON

Table 172. Bit Descriptions for I2C0SHCON

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0000	RW
0	RESET	Write a 1 to this bit to reset the I²C start and stop detection circuits. Setting this bit resets the LINEBUSY status bit.	0x0	W

Automatic Stretch Control Register

Address: 0x40003058, Reset: 0x0000, Name: I2C0ASSCL

Table 173. Bit Descriptions for I2C0ASSCL

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
9	SSRTSTA	Stretch timeout status bit for slave. Asserts when slave automatic stretch mode has timed out. Cleared when this bit is read.	0x0	R
8	MSRTSTA	Stretch timeout status bit for master. Asserts when master automatic stretch mode has timed out. Cleared when this bit is read.	0x0	R
[7:4]	SSTRCON	Automatic stretch mode control for the slave. These bits control automatic stretch mode for slave operation. Slave stretch control allows the slave to hold the SCL line low and to gain more time to service an interrupt, load a FIFO, or read a FIFO. Use the timeout feature to avoid a bus lockup condition where the slave indefinitely holds SCL low. As a slave transmitter, SCL is automatically stretched from the negative edge of SCL if the slave Tx FIFO is empty before sending acknowledge/no acknowledge for address byte, or before sending data for a data byte. Stretching stops when the slave Tx FIFO is no longer empty or if a timeout occurs. As a slave receiver, the SCL clock is automatically stretched from the negative edge of SCL, when the slave Rx FIFO is full, before sending acknowledge/no acknowledge. Stretching stops when slave Rx FIFO is no longer in an overflow condition or if a timeout occurs. 0000: automatic slave clock stretching disabled. 0001 to 1110: automatic slave clock stretching enabled. The timeout period is defined by the following equation: $\frac{(\mathrm{I2CDIV}[15:8]+\mathrm{I2CDIV}[7:0])-1}{(\mathrm{UCLK}/\mathrm{CLKCON}[10:8])}\times\left(2^{\mathrm{I2CASSCL}[7:4]}\right)$ The I²C bus baud rate has no influence on slave stretch timeout period. 1111: automatic slave clock stretching enabled with indefinite timeout period.	0x0	RW
[3:0]	MSTRCON	Automatic stretch mode control for the master. These bits control automatic stretch mode for master operation. Master stretch control allows master to hold the SCL line low and gain more time to service an interrupt, load a FIFO or read a FIFO. Use the timeout feature to avoid a bus lockup condition where the master indefinitely holds SCL low. As a master transmitter, SCL is automatically stretched from the negative edge of SCL if the master Tx FIFO is empty before sending acknowledge/no acknowledge for address byte, or before sending data for a data byte. Stretching stops when the master Tx FIFO is no longer empty or if a timeout occurs. As a master receiver, the SCL clock is automatically stretched from the negative edge of SCL, when the master Rx FIFO is full, before sending acknowledge/no acknowledge. Stretching stops when the master Rx FIFO is no longer in an overflow condition or if a timeout occurs. 0000: automatic master clock stretching disabled 0001 to 1110: automatic master clock stretching enabled. The timeout period is defined by the following equation: $\frac{(\mathrm{I2CDIV}[15:8]+\mathrm{I2CDIV}[7:0]-1)}{(\mathrm{UCLK}/\mathrm{CLKCON}[10:8])}\times\left(2^{\mathrm{I2SCASSCL}[3:0]}\right)$ 1111: automatic master clock stretching enabled with indefinite time-out period.	0x0	RW

REGISTER SUMMARY: I^{2}C1

Table 174. I^{2}C1 Register Summary

Address	Name	Description	Reset	Access
0x40003400	I2C1MCON	Master control register	0x0000	RW
0x40003404	I2C1MSTA	Master status register	0x6000	R
0x40003408	I2C1MRX	Master receive data register	0x0000	R
0x4000340C	I2C1MTX	Master transmit data register	0x0000	RW
0x40003410	I2C1MRXCNT	Master receive data count register	0x0000	RW
0x40003414	I2C1MCRXCNT	Master current receive data count register	0x0000	R
0x40003418	I2C1ADR0	First master address byte register	0x0000	RW
0x4000341C	I2C1ADR1	Second master address byte register	0x0000	RW
0x40003424	I2C1DIV	Serial clock period divisor register	0x1F1F	RW
0x40003428	I2C1SCON	Slave control register	0x0000	RW
0x4000342C	I2C1SSTA	Slave I^{2}C status/error/IRQ register	0x0001	R
0x40003430	I2C1SRX	Slave receive register	0x0000	R
0x40003434	I2C1STX	Slave transmit register	0x0000	RW
0x40003438	I2C1ALT	Hardware general call ID register	0x0000	RW
0x4000343C	I2C1ID0	First slave address device ID register	0x0000	RW
0x40003440	I2C1ID1	Second slave address device ID register	0x0000	RW
0x40003444	I2C1ID2	Third slave address device ID register	0x0000	RW
0x40003448	I2C1ID3	Fourth slave address device ID register	0x0000	RW
0x4000344C	I2C1FSTA	Master and slave FIFO status register	0x0000	RW
0x40003450	I2C1SHCON	Master and slave shared control register	0x0000	W
0x40003458	I2CASSCL	Automatic stretch control register for master and slave mode	0x0000	RW

REGISTER DETAILS: I^{2}C1

Master Control Register

Address: 0x40003400, Reset: 0x0000, Name: I2C1MCON

Table 175. Bit Descriptions for I2C1MCON

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
11	MTXDMA	Enable master Tx DMA request. 0: disable DMA mode. 1: enable I^{2}C master DMA Tx requests.	0x0	W
10	MRXDMA	Enable master Rx DMA request. 0: disable DMA mode. 1: enable I^{2}C master DMA Rx requests.	0x0	W
9	RESERVED	Reserved.	0x0	RW
8	IENCMP	Transaction completed (or stop detected) interrupt enable. 0: an interrupt is not generated when a stop is detected. 1: an interrupt is generated when a stop is detected.	0x0	RW
7	IENACK	Acknowledge not received interrupt enable. 0: disable acknowledge not received interrupt. 1: enable acknowledge not received interrupt.	0x0	RW
6	IENALOST	Arbitration lost interrupt enable. 0: disable arbitration lost interrupt. 1: enable arbitration lost interrupt.	0x0	RW
5	IENMTX	Transmit request interrupt enable. 0: disable transmit request interrupt. 1: enable transmit request interrupt.	0x0	RW
4	IENMRX	Receive request interrupt enable. 0: disable receive request interrupt. 1: enable receive request interrupt.	0x0	RW
3	STRETCH	Stretch SCL enable. 0: disable Clock stretching. 1: setting this bit tells the device if SCL is 0, hold it at 0; or if SCL is 1, when it next goes to 0, hold it at 0.	0x0	RW
2	LOOPBACK	Internal loopback enable. Note that is also possible for the master to loop back a transfer to the slave as long as the device address corresponds, that is, external loopback. 0: SCL and SDA out of the device are not muxed onto their corresponding inputs. 1: SCL and SDA out of the device are muxed onto their corresponding inputs.	0x0	RW
1	COMPETE	Start backoff disable. Setting this bit enables the device to compete for ownership even if another device is currently driving a start condition.	0x0	RW
0	MASEN	Master enable. Disable the master when not in use to gate the clock to the master and save power. Do not clear this bit until a transaction has completed; see the TCOMP bit in the master status register. 0: master is disabled. 1: master is enabled.	0x0	RW

Master Status Register

Address: 0x40003404, Reset: 0x6000, Name: I2C1MSTA

Table 176. Bit Descriptions for I2C1MSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	SCL_FILTERED	State of SCL line. This bit is the output of the glitch filter on SCL. SCL is always pulled high when undriven.	0x1	R
13	SDA_FILTERED	State of SDA line. This bit is the output of the glitch-filter on SDA. SDA is always pulled high when undriven.	0x1	R
12	MTXUFLOW	Master transmit underflow. MTXUFLOW asserts when the I^{2}C master ends the transaction due to a Tx FIFO empty condition. This bit is asserted only when the IENMTX bit is set.	0x0	RC
11	MSTOP	Stop driven by this I^{2}C master. MSTOP asserts when this I^{2}C master drives a stop condition on the I^{2}C bus. This bit, when asserted, can indicate a transaction completion, Tx underflow, Rx overflow, or a no acknowledge by the slave. This is different from TCOMP because this bit is not asserted when the stop condition occurs due to any other I^{2}C master. No interrupt is generated for the assertion of this bit. However, if IENCMP is 1, every stop condition generates an interrupt and this bit can be read. When this bit is read, it clears status.	0x0	RC
10	LINEBUSY	Line is busy. This bit asserts when a start is detected on the I^{2}C bus. This bit deasserts when a stop is detected on the I^{2}C bus.	0x0	R
9	MRXOF	Master receive FIFO overflow. This bit asserts when a byte is written to the receive FIFO when the FIFO is already full. When the bit is read, it clears the status.	0x0	RC
8	TCOMP	Transaction complete or stop detected. This bit asserts when a stop condition is detected on the I^{2}C bus. If IENCMP is 1, an interrupt is generated when this bit asserts. This bit only asserts if the master is enabled (MASEN = 1). Use this bit to determine when it is safe to disable the master. It can also wait for another master transaction to complete on the I^{2}C bus when this master loses arbitration. When this bit is read, it clears status. This bit can drive an interrupt.	0x0	RC
7	NACKDATA	Acknowledge not received in response to data write. This bit asserts when an acknowledge is not received in response to a data write transfer. If IENACK is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt. This bit is cleared on a read of the I2C1MSTA register.	0x0	RC
6	MBUSY	Master busy. This bit indicates that the master state machine is servicing a transaction. It is clear if the state machine is idle or if another device has control of the I^{2}C bus.	0x0	R
5	ALOST	Arbitration lost. This bit asserts if the master loses arbitration. If IENALOST is 1, an interrupt is generated when this bit asserts. This bit is cleared on a read of the I2C1MSTA register. This bit can drive an interrupt.	0x0	RC
4	NACKADDR	Acknowledge not received in response to an address. This bit asserts if an acknowledge is not received in response to an address. If IENACK is 1, an interrupt is generated when this bit asserts. This bit is cleared on a read of the I2CMSTA register. This bit can drive an interrupt.	0x0	RC
3	MRXREQ	Master receive request. This bit asserts when there is data in the receive FIFO. If IENMRX is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt.	0x0	R
2	MTXREQ	Master transmit request. This bit asserts when the direction bit is 0 and the transmit FIFO is either empty or not full. If IENMTX is 1, an interrupt is generated when this bit asserts. This bit can drive an interrupt.	0x0	R
[1:0]	MTXFSTA	Master transmit FIFO status. These two bits show the master transmit FIFO status and can be decoded as follows: 00 = FIFO empty. 10 = 1 byte in FIFO. 11 = FIFO full.	0x0	R

Master Receive Data Register

Address: 0x40003408, Reset: 0x0000, Name: I2C1MRX

Table 177. Bit Descriptions for I2C1MRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ICMRX	Master receive register. This register allows access to the receive data FIFO. The FIFO can hold two bytes.	0x0	R

Master Transmit Data Register

Address: 0x4000340C, Reset: 0x0000, Name: I2C1MTX

Table 178. Bit Descriptions for I2C1MTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	I2CMTX	Master transmit register. For test and debug purposes, when read, this register returns the byte that is currently being transmitted by the master. That is, a byte written to the transmit register can be read back some time later when that byte is being transmitted on the line. This register allows access to the transmit data FIFO. The FIFO can hold two bytes.	0x0	RW

Master Receive Data Count Register

Address: 0x40003410, Reset: 0x0000, Name: I2C1MRXCNT

Table 179. Bit Descriptions for I2C1MRXCNT

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	EXTEND	Extended read. Use this bit if more than 256 bytes are required on a read. For example, to receive 412 bytes, write 0x100 (EXTEND = 1) to the I2CMRXCNT register. Wait for the first byte to be received, then check the I2CMCRXCNT register for every byte received thereafter. When COUNT returns to 0, 256 bytes have been received. Then write 0x09C to the I2CMRXCNT register.	0x0	RW
[7:0]	COUNT	Receive count. Program the number of bytes required minus one to this register. If just 1 byte is required, write 0 to this register. If more than 256 bytes are required, use EXTEND.	0x0	RW

Master Current Receive Data Count Register

Address: 0x40003414, Reset: 0x0000, Name: I2C1MCRXCNT

Table 180. Bit Descriptions for I2C1MCRXCNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	COUNT	Current receive count. This register gives the total number of bytes received so far. If 256 bytes are requested, this register reads 0 when the transaction has completed.	0x0	R

First Master Address Byte Register

Address: 0x40003418, Reset: 0x0000, Name: I2C1ADR0

Table 181. Bit Descriptions for I2C1ADR0

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ADR0	Address Byte 0. If a 7-bit address is required, Bit 7 to Bit 1 of ADR0 are programmed with the address and Bit 0 of ADR0 is programmed with the direction (0 = write, 1 = read). If a 10 bit address is required, Bit 7 to Bit 3 of ADR0 are programmed with 11110, Bit 2 to Bit 1 of ADR0 are programmed with the 2 MSBs of the address, and Bit 0 of ADR0 is programmed to 0.	0x0	RW

Second Master Address Byte Register

Address: 0x4000341C, Reset: 0x0000, Name: I2C1ADR1

Table 182. Bit Descriptions for I2C1ADR1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ADR1	Address Byte 1. This register is only required when addressing a slave with a 10-bit address. Bit 7 to Bit 0 of ADR1 are programmed with the lower 8 bits of the address.	0x0	RW

Serial Clock Period Divisor Register

Address: 0x40003424, Reset: 0x1F1F, Name: I2C1DIV

Table 183. Bit Descriptions for I2C1DIV

Bits	Bit Name	Description	Reset	Access
[15:8]	HIGH	Serial clock high time. This register controls the clock high time. The timer is driven by the core clock (UCLK). Use the following equation to derive the required high time. *HIGH* = (*REQD_HIGH_TIME*/*UCLK_PERIOD*) − 2 For example, to generate a 400 kHz SCL with a low time of 1300 ns and a high time of 1200 ns, with a core clock frequency of 50 MHz: *LOWTIME* = 1300 ns/20 ns − 1 = 0x40 (64 decimal) *HIGH* = 1200 ns/20 ns − 2 = 0x3A (58 decimal) This register is reset to 0x1F, which gives an SCL high time of 33 UCLK ticks. $t_{HD:STA}$ is also determined by the HIGH value. $t_{HD:STA} = (HIGH - 1) \times UCLK\_PERIOD$ As $t_{HD:STA}$ must be 600 ns, with UCLK = 50 MHz, the minimum value for HIGH is 31. This gives an SCL high time of 660 ns.	0x1F	RW
[7:0]	LOW	Serial clock low time. This register controls the clock low time. The timer is driven by the UCLK. Use the following equation to derive the required low time. *LOW* = (*REQD_LOW_TIME*/*UCLK_PERIOD*) − 1 This register is reset to 0x1F, which gives an SCL low time of 32 UCLK ticks.	0x1F	RW

Slave Control Register

Address: 0x40003428, Reset: 0x0000, Name: I2C1SCON

Table 184. Bit Descriptions for I2C1SCON

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	STXDMA	Enable slave Tx DMA request. Set to 1 by user code to enable I^{2}C slave DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	RW
13	SRXDMA	Enable slave Rx DMA request. Set to 1 by user code to enable I^{2}C slave DMA Rx requests. Cleared by user code to disable DMA mode.	0x0	RW
12	IENREPST	Repeated start interrupt enable. If 1, an interrupt is generated when the REPSTART status bit asserts. If 0, an interrupt is not generated when the REPSTART status bit asserts.	0x0	RW
11	SXMITDEC	Decrement slave Tx FIFO status when a byte has been transmitted. If set to 1, the transmit FIFO status is decremented when a byte has been transmitted. If set to 0, the transmit FIFO status is decremented when the byte is unloaded from the FIFO into a shadow register at the start of byte transmission.	0x0	RW
10	IENSTX	Slave transmit request interrupt enable.	0x0	RW
9	IENSRX	Slave receive request interrupt enable.	0x0	RW
8	IENSTOP	Stop condition detected interrupt enable.	0x0	RW
7	NACK	No acknowledge next communication. If this bit is set, the next communication is not acknowledged. This can be used if during a 24xx style access, an attempt was made to write to a read only or nonexistent location in system memory. That is the indirect address in a 24xx style write pointed to an unwritable memory location.	0x0	RW
6	STRETCHSCL	Stretch SCL enable. Setting this bit tells the device if SCL is 0, hold it at 0; or if SCL is 1, when it next goes to 0, hold it at 0.	0x0	RW
5	EARLYTXR	Early transmit request mode. Setting this bit enables a transmit request just after the positive edge of the direction bit SCL clock pulse.	0x0	RW
4	GCSBCLR	General call status bit clear. The general call status and general call ID bits are cleared when a 1 is written to this bit. The general call status and general call ID bits are not reset by anything other than a write to this bit or a full reset.	0x0	W
3	HGCEN	Hardware general call enable. When this bit and the general call enable bit are set, the device after receiving a general call, Address 00h, and a data byte checks the contents of the ALT against the receive shift register. If they match, the device has received a hardware general call. If a device requires urgent attention from a master device without knowing which master it requires to turn to, it can use this call. This is a call "to whom it may concern". The device that requires attention embeds its own address into the message. The LSB of the ALT register must always be written to a 1, as per I^{2}C January 2000 specification.	0x0	RW
2	GCEN	General call enable. This bit enables the I^{2}C slave to acknowledge an I^{2}C general call, Address 0x00 (write).	0x0	RW
1	ADR10EN	Enabled 10-bit addressing. If this bit is clear, the slave can support four slave addresses, programmed in Register I2CID0 to Register I2CID3. When this bit is set, 10-bit addressing is enabled. One 10-bit address is supported by the slave and is stored in I2CID0 and I2CID1, where I2CID0 contains the first byte of the address and the upper five bits must be programmed to 11110. I2CID3 and I2CID4 can be programmed with 7-bit addresses at the same time.	0x0	RW
0	SLVEN	Slave enable. When 1, the slave is enabled. When 0, all slave state machine flops are held in reset, and the slave is disabled.	0x0	RW

Slave I^{2}C Status/Error/IRQ Register

Address: 0x4000342C, Reset: 0x0001, Name: I2C1SSTA

Table 185. Bit Descriptions for I2C1SSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	START	Start and matching address. This bit is asserted if a start is detected on SCL/SDA and the device address matched, or a general call (address = 0000_0000) code is received and general call is enabled; or if a high speed (address = 0000_1XXX) code is received; or if a start byte (0000_0001) is received. It is cleared on receipt of either a stop or start condition.	0x0	R
13	REPSTART	Repeated start and matching address. This bit is asserted if start is already asserted and then a repeated start is detected. It is cleared when read or on receipt of a stop condition. This bit can drive an interrupt.	0x0	RC
[12:11]	IDMAT	Device ID matched. 00: received address matched ID Register 0. 01: received address matched ID Register 1. 10: received address matched ID Register 2. 11: received address matched ID Register 3.	0x0	R
10	STOP	Stop after start and matching address. This bit is set by hardware if the slave device received a stop condition after a previous start condition and a matching address. It is cleared by a read of the status register. If STOPINTEN in the slave control register is asserted, the slave interrupt request asserts when this bit is set. This bit can drive an interrupt.	0x0	RC
[9:8]	GCID	General ID. GCID is cleared when the GCSBCLR is written to 1. These status bits are not cleared by a general call reset. 00: no general call. 01: general call reset and program address. 10: general call program address. 11: general call matching alternative ID.	0x0	R
7	GCINT	General call interrupt. This bit always drives an interrupt. The bit is asserted if the slave device receives a general call of any type. To clear, write 1 to the GCSBCLR in the slave control register. If it was a general call reset, all registers are at their default values. If it was a hardware general call, the Rx FIFO holds the second byte of the general call, and this can be compared with the ALT register.	0x0	R
6	SBUSY	Slave busy. This bit is set by hardware if the slave device receives an I^{2}C start condition. It is cleared by hardware when the address does not match an ID register, when the slave device receives a I^{2}C stop condition, or if a repeated start address does not match.	0x0	R
5	NACK	Acknowledge not generated by the slave. When asserted, this bit indicates that the slave responded to its device address with a no acknowledge. It is asserted if there was no data to transmit and the sequence was a slave read, or if the no acknowledge bit was set in the slave control register and the device was addressed. This bit is cleared on a read of the I2CSSTA register.	0x0	RC
4	SRXOF	Slave receive FIFO overflow. This bit asserts when a byte is written to the slave receive FIFO when the FIFO is already full.	0x0	RC
3	SRXREQ	Slave receive request. SRXREQ asserts whenever the slave receive FIFO is not empty. Read or flush the slave receive FIFO to clear this bit. This bit asserts on the falling edge of the SCL clock pulse that clocks in the last data bit of a byte. This bit can drive an interrupt.	0x0	RC
2	STXREQ	Slave transmit request. If EARLYTXR = 0, STXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted on the negative edge of the SCL pulse that clocks in the direction bit (if the device address matched also). If EARLYTXR = 1, STXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted after the positive edge of the SCL pulse that clocks in the direction bit (if the device address matched also). This bit is cleared on a read of the I2CSSTA register.	0x0	RC

Bits	Bit Name	Description	Reset	Access
1	STXUR	Slave transmit FIFO underflow. This bit is set if a master requests data from the device, and the Tx FIFO is empty for the rising edge of SCL.	0x0	RC
0	STXFSEREQ	Slave Tx FIFO status or early request. If EARLYTXR = 0, this bit is asserted whenever the slave Tx FIFO is empty. If EARLYTXR = 1, TXFSEREQ is set when the direction bit for a transfer is received high. It asserts on the positive edge of the SCL clock pulse that clocks in the direction bit (if the device address also matches). It only asserts once for a transfer. It is cleared when read if EARLYTXR is asserted.	0x1	RW

Slave Receive Register

Address: 0x40003430, Reset: 0x0000, Name: I2C1SRX

Table 186. Bit Descriptions for I2C1SRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved	0x0	R
[7:0]	I2CSRX	Slave receive register	0x0	R

Slave Transmit Register

Address: 0x40003434, Reset: 0x0000, Name: I2C1STX

Table 187. Bit Descriptions for I2C1STX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved	0x0	R
[7:0]	I2CSTX	Slave transmit register	0x0	RW

Hardware General Call ID Register

Address: 0x40003438, Reset: 0x0000, Name: I2C1ALT

Table 188. Bit Descriptions for I2C1ALT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ALT	Slave alt. This register is used in conjunction with I2CSCON[3] to match a master generating a hardware general call. It is used in the case where a master device cannot be programmed with a slave address and instead the slave must recognize the master address.	0x0	RW

First Slave Address Device ID Register

Address: 0x4000343C, Reset: 0x0000, Name: I2C1ID0

Table 189. Bit Descriptions for I2C1ID0

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID0	Slave device ID 0. I2CID0[7:1] is programmed with the device ID. I2CID0[0] is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Second Slave Address Device ID Register

Address: 0x40003440, Reset: 0x0000, Name: I2C1ID1

Table 190. Bit Descriptions for I2C1ID1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID1	Slave device ID 1. I2CID1[7:1] is programmed with the device ID. I2CID1[0] is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Third Slave Address Device ID Register

Address: 0x40003444, Reset: 0x0000, Name: I2C1ID2

Table 191. Bit Descriptions for I2C1ID2

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID2	Slave device ID 2. I2CID2[7:1] is programmed with the device ID. I2CID2[0] is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Fourth Slave Address Device ID Register

Address: 0x40003448, Reset: 0x0000, Name: I2C1ID3

Table 192. Bit Descriptions for I2C1ID3

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	ID3	Slave device ID 3. I2CID3[7:1] is programmed with the device ID. I2CID3[0] is don't care. See the ADR10EN bit in the slave control register to see how this register is programmed with a 10-bit address.	0x0	RW

Master and Slave FIFO Status Register

Address: 0x4000344C, Reset: 0x0000, Name: I2C1FSTA

Table 193. Bit Descriptions for I2C1FSTA

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	RW
9	MFLUSH	Flush the master transmit FIFO. 0: clearing to 0 has no effect. 1: set to 1 to flush the master transmit FIFO. The master transmit FIFO must be flushed if arbitration is lost or if a slave responds with a no acknowledge.	0x0	W
8	SFLUSH	Flush the slave transmit FIFO. 0: clearing to 0 has no effect. 1: set to 1 to flush the slave transmit FIFO.	0x0	W
[7:6]	MRXFSTA	Master receive FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R
[5:4]	MTXFSTA	Master transmit FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R
[3:2]	SRXFSTA	Slave receive FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R
[1:0]	STXFSTA	Slave transmit FIFO status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty. 01: 1 byte in the FIFO. 10: 2 bytes in the FIFO. 11: reserved.	0x0	R

Master and Slave Shared Control Register

Address: 0x40003450, Reset: 0x0000, Name: I2C1SHCON

Table 194. Bit Descriptions for I2C1SHCON

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0000	RW
0	RESET	Write a 1 to this bit to reset the I²C start and stop detection circuits. Setting this bit resets the LINEBUSY status bit.	0x0	W

Automatic Stretch Control Register

Address: 0x40003458, Reset: 0x0000, Name: I2C1ASSCL

Table 195. Bit Descriptions for I2C1ASSCL

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x0	R
9	SSRTSTA	Stretch timeout status bit for slave. This bit asserts when slave automatic stretch mode has timed out. It is cleared when this bit is read.	0x0	R
8	MSRTSTA	Stretch timeout status bit for master. This bit asserts when master automatic stretch mode has timed out. It is cleared when this bit is read.	0x0	R
[7:4]	SSTRCON	Automatic stretch mode control for slave. These bits control automatic stretch mode for slave operation. Slave stretch control allows the slave to hold the SCL line low and gain more time to service an interrupt, load a FIFO, or read a FIFO. Use the timeout feature to avoid a bus lockup condition where the slave indefinitely holds SCL low. As a slave transmitter, SCL is automatically stretched from the negative edge of SCL if the slave Tx FIFO is empty before sending acknowledge/no acknowledge for address byte, or before sending data for a data byte. Stretching stops when the slave Tx FIFO is no longer empty or if a timeout occurs. As a slave receiver, SCL clock is automatically stretched from the negative edge of SCL, when the slave Rx FIFO is full, before sending acknowledge/no acknowledge. Stretching stops when the slave Rx FIFO is no longer in an overflow condition or if a timeout occurs. 0000: automatic slave clock stretching disabled. 0001 to 1110: automatic slave clock stretching enabled. The timeout period is defined by the following equation: $\frac{(\text{I2CDIV}[15:8]+\text{I2CDIV}[7:0])-1}{(\text{UCLK}/\text{CLKCON}[10:8])}\times\left(2^{\text{I2CASSCL}[7:4]}\right)$ The I²C bus baud rate has no influence on the slave stretch timeout period. 1111: automatic slave clock stretching enabled with indefinite timeout period	0x0	RW
[3:0]	MSTRCON	Automatic stretch mode control for master. These bits control automatic stretch mode for master operation. Master stretch control allows master to hold the SCL line low and gain more time to service an interrupt, load a FIFO, or read a FIFO. Use the timeout feature to avoid a bus lockup condition where the master indefinitely holds SCL low. As a master transmitter, SCL is automatically stretched from the negative edge of SCL if the master Tx FIFO is empty before sending acknowledge/no acknowledge for address byte, or before sending data for a data byte. Stretching stops when the master Tx FIFO is no longer empty or if a timeout occurs. As a master receiver, SCL clock is automatically stretched from the negative edge of SCL, when the master Rx FIFO is full, before sending acknowledge/no acknowledge. Stretching stops when the master Rx FIFO is no longer in an overflow condition or if a timeout occurs. 0000: automatic master clock stretching disabled. 0001 to 1110: automatic master clock stretching enabled. The timeout period is defined by the following equation: $\frac{(\text{I2CDIV}[15:8]+\text{I2CDIV}[7:0])-1}{(\text{UCLK}/\text{CLKCON}[10:8])}\times\left(2^{\text{I2CASSCL}[3:0]}\right)$ 1111: automatic master clock stretching enabled with indefinite timeout period.	0x0	RW

SERIAL PERIPHERAL INTERFACES

SPI FEATURES

The ADuCM310 integrates two complete hardware serial peripheral interfaces with the following standard SPI features:

- Serial clock phase mode and serial clock polarity mode
- LSB first transfer option
- Loopback mode
- Master or slave mode
- Transfer and interrupt mode
- Continuous transfer mode
- Tx/Rx FIFO
- Interrupt mode, interrupt after one, two, three, or four bytes
- Rx overflow mode and Tx underrun mode
- Open-circuit data output mode
- Full-duplex communications supported (simultaneous transmit/receive)

SPI OVERVIEW

The ADuCM310 integrates two complete hardware serial peripheral interfaces (SPI). SPI is an industry standard, synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, that is, full duplex. The two SPIs implemented on the ADuCM310 can operate to a maximum bit rate of 20 Mbps in both master and slave mode.

The SPI blocks have an additional DMA feature. It has two DMA channels that interface with a µDMA controller of the ARM Cortex-M3 processor. One DMA channel is used for transmit and the other for receive.

SPI OPERATION

The SPI port can be configured for master or slave operation and consists of four pins: MISO, MOSI, SCLK, and $\overline{\text{CS}}$.

Note that the GPIOs used for SPI communication must be configured in SPI mode before enabling the SPI peripheral, and that the internal pull-up resistors on the SPI pins must be disabled via the GPxPUL registers when using the SPI.

MISO (Master In, Slave Out) Pin

The MISO pin is configured as an input line in master mode and an output line in slave mode. The MISO line on the master (data in) must be connected to the MISO line in the slave device (data out). The data is transferred as byte-wide (8-bit) serial data, MSB first.

MOSI (Master Out, Slave In) Pin

The MOSI pin is configured as an output line in master mode and an input line in slave mode. The MOSI line on the master (data out) must be connected to the MOSI line in the slave device (data in). The data is transferred as byte-wide (8-bit) serial data, MSB first.

SCLK (Serial Clock Input/Output) Pin

The master serial clock (SCLK) synchronizes the data being transmitted and received through the MOSI SCLK period. Therefore, a byte is transmitted/received after eight SCLK periods. The SCLK pin is configured as an output in master mode and as an input in slave mode.

In master mode, the polarity and phase of the clock are controlled by the SPIxCON register, and the bit rate is defined in the SPIxDIV register as follows:

$$f_{SERIALCLOCK} = \frac{SPICLK}{2 \times (1 + SPIxDIV)}$$

where *SPICLK* is the 80 MHz system clock divided by the factor set in CLKCON1[2:0].

It is possible to disable the clocks to SPI0 and SPI1 separately:

- CLKCON1[0] = 1 disables the clock to SPI0.
- CLKCON1[1] = 1 disables the clock to SPI1.

By reducing the clock rate to the SPI blocks, it is possible to reduce the power consumption of the SPI block. The maximum data rate is 20 Mbps.

In slave mode, the SPIxCON register must be configured with the phase and polarity of the expected input clock. The slave accepts data from an external master at rates of up to 20 Mbps.

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

Chip Select ($\overline{\text{CS}}$ Input) Pin

In SPI slave mode, a transfer is initiated by the assertion of $\overline{\text{CS}}$, which is an active low input signal. The SPI port then transmits and receives 8-bit data until the transfer is concluded by the deassertion of $\overline{\text{CS}}$. In slave mode, $\overline{\text{CS}}$ is always an input.

In SPI master mode, $\overline{\text{CS}}$ is an active low output signal. It asserts itself automatically at the beginning of a transfer and deasserts itself upon completion.

$\overline{\text{CS}}$ must always be configured as an SPI pin in GPxCON when the SPI is used. If an ADuCM310 master wants to communicate with multiple SPI slaves, $\overline{\text{CS}}$ must be left floating, and the GPIOs can be connected to the $\overline{\text{CS}}$ lines of the slaves. The CSRSG and CSFLG bits in the SPIxSTA register (Bits[14:13]) can determine the right time to bring the GPIOs low and high.

SPI TRANSFER INITIATION

In master mode, the transfer and interrupt mode bit, TIM (SPIxCON[6]), determines the manner in which an SPI serial transfer is initiated. If the TIM bit is set, a serial transfer is initiated after a write to the Tx FIFO. If the TIM bit is cleared, a serial transfer is initiated after a read of the Rx FIFO; the read must be performed while the SPI interface is idle. A read performed during an active transfer does not initiate another transfer.

For any setting of SPIxCON[1] and SPIxCON[6], the SPI simultaneously receives and transmits data. Therefore, during data transmission, the SPI is also receiving data and filling up the Rx FIFO. If the data is not read from the Rx FIFO, the overflow interrupt occurs when the FIFO starts to overflow. If the user does not want to read the Rx data or receive overflow interrupts, SPIxCON[12] can be set, and the receive data is not saved to the Rx FIFO.

Similarly, when the user wants to only receive data and does not want to write data to the Tx FIFO, SPIxCON[13] can be set to avoid receiving underrun interrupts from the Tx FIFO.

Tx Initiated Transfer

For transfers initiated by a write to the Tx FIFO, the SPI starts transmitting as soon as the first byte is written to the FIFO, irrespective of the configuration in SPIxCON[15:14]. The first byte is immediately read from the FIFO, written to the Tx shift register, and the transfer commences.

If the continuous transfer enable bit, SPIxCON[11], is set, the transfer continues until no valid data is available in the Tx FIFO. There is no stall period between transfers where $\overline{\text{CS}}$ is deasserted; $\overline{\text{CS}}$ is asserted and remains asserted for the duration of the transfer until the Tx FIFO is empty. Determining when the transfer stops does not depend on SPIxCON[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, SPIxCON[11], is cleared, each transfer consists of a single 8-bit serial transfer. If valid data exists in the Tx FIFO, a new transfer is initiated after a stall period where $\overline{\text{CS}}$ is deasserted.

Rx Initiated Transfer

Transfers initiated by a read of the Rx FIFO depend on the number of bytes to be received in the FIFO. If SPIxCON[15:14] = 11 and a read to the Rx FIFO occurs, the SPI initiates a 4-byte transfer. If continuous mode is set, the four bytes occur continuously with no deassertion of $\overline{\text{CS}}$ between bytes. If continuous mode is not set, the four bytes occur with stall periods between transfers where the $\overline{\text{CS}}$ is deasserted. A read of the Rx FIFO while the SPI is receiving data does not initiate another transfer after the present transfer is complete.

In slave mode, a transfer is initiated by the assertion of $\overline{\text{CS}}$ ($\overline{\text{CS}}$ = 0).

The device as a slave transmits and receives 8-bit data until the transfer is concluded by the deassertion of $\overline{\text{CS}}$ ($\overline{\text{CS}}$ = 1).

The SPI transfer protocol diagrams (see Figure 24 and Figure 25) illustrate the data transfer protocol for the SPI and the effects of the CPHA and CPOL bits in the control register (SPIxCON) on that protocol.

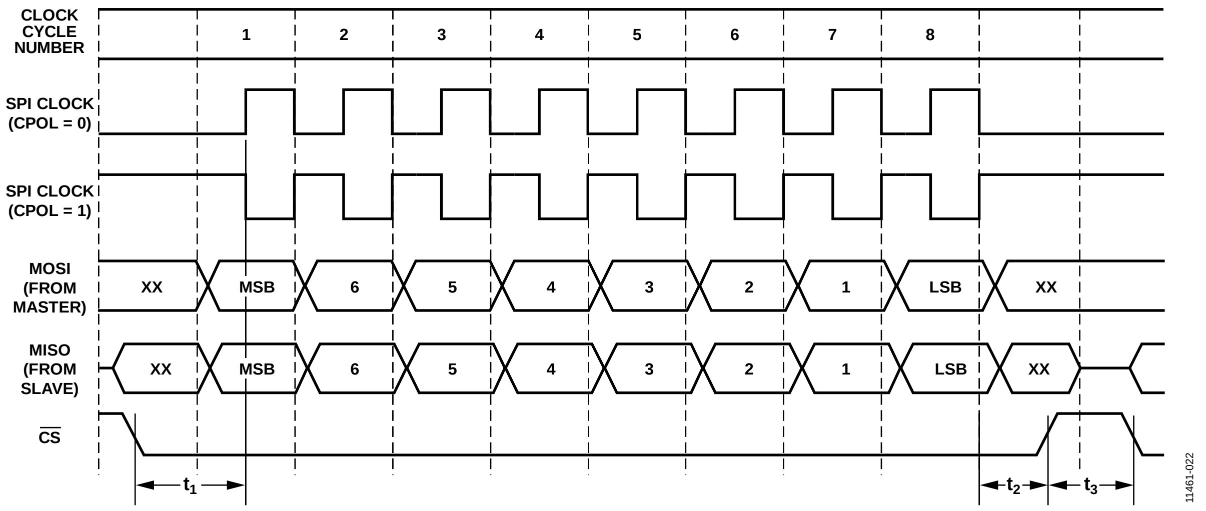


Figure 24. SPI Transfer Protocol, CPHA = 0

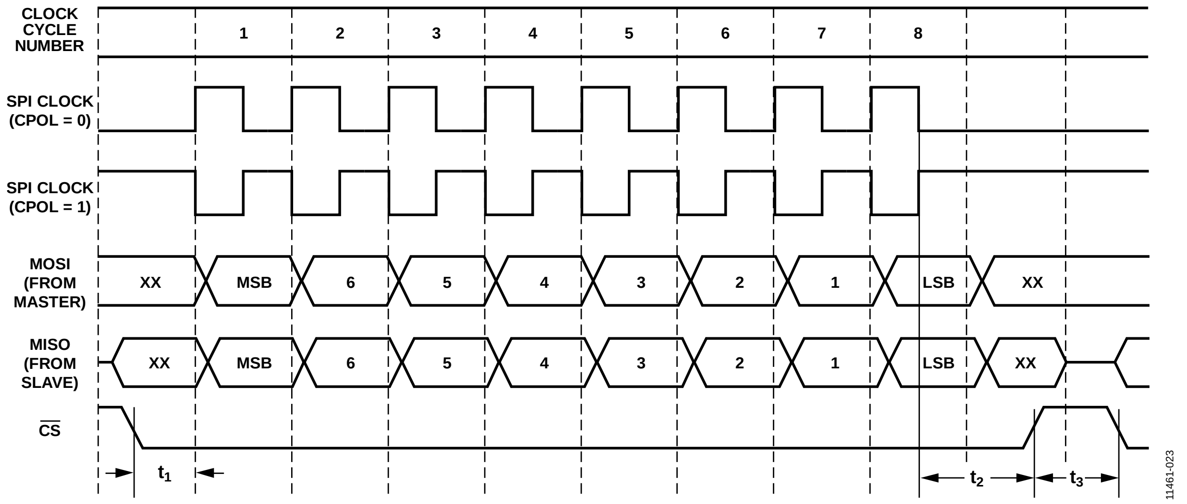


Figure 25. SPI Transfer Protocol, CPHA = 1

SPI Data Underrun and Overflow

If the transmit zeros enable bit, ZEN (SPIxCON[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 to 1, 0s are transmitted when a transfer is initiated with no valid data in the FIFO.

If the Rx overflow overwrite enable bit, RXOF(SPIxCON[8]), is set, the valid data in the Rx FIFO is overwritten by the new serial byte received if there is no space left in the FIFO. If RXOF is cleared, the new serial byte received is discarded if there is no space left in the FIFO.

When the RXOF is set, the contents of the SPI Rx FIFO are undefined, and its contents must be discarded by user code.

Full-Duplex Operation

Simultaneous read/writes are supported on the SPI.

When implementing full-duplex transfers in master mode, use the following procedure:

1. Initiate a transfer sequence via a transmit on the MOSI pin. Set SPIxCON[6] = 1. If interrupts are enabled, interrupts are triggered when a transmit interrupt occurs but not when a byte is received.
2. If using interrupts, the SPI Tx interrupt indicated by SPIxSTA[5] or the Tx FIFO underrun interrupt (SPIxSTA[4]) is asserted approximately 3 × SPICLK to 4 × SPICLK periods into the transfer of the first byte. Reload a byte into the Tx FIFO, if necessary, by writing to SPIxTX.
3. The first byte received via the MISO pin does not update the Rx FIFO status bits (SPIxSTA[10:8]) until 12 × SPICLK periods after $\overline{\text{CS}}$ goes low. Therefore, two transmit interrupts may occur before the first receive byte is ready to be handled.
4. After the last transmit interrupt occurs, it may be necessary to read two more bytes. It is recommended that SPIxSTA[10:8] be polled outside of the SPI interrupt handler after the last transmit interrupt is handled.

SPI INTERRUPTS

There is one interrupt line per SPI and four sources of interrupts. SPIxSTA[0] reflects the state of the interrupt line, and SPIxSTA[7:4] reflects the state of the four sources.

The SPI generates either TIRQ or RIRQ. Both interrupts cannot be enabled at the same time. The appropriate interrupt is enabled using the TIM bit, SPIxCON[6]. If TIM = 1, TIRQ is enabled. If TIM = 0, RIRQ is enabled.

Also note that the SPI0 and SPI1 interrupt source must be enabled in the NVIC register, ISER0 (ISER0[19] = SPI0, ISER0[20] = SPI1).

Tx Interrupt

If TIM (SPIxCON[6]) is set, the Tx FIFO status causes the interrupt. SPIxCON[15:14] control when the interrupt occurs, as shown in Table 196.

Table 196. SPIxCON[15:14] IRQ Mode Bits

SPIxCON[15:14]	Interrupt Condition
00	An interrupt is generated after each byte that is transmitted. The interrupt occurs when the byte is read from the FIFO and written to the shift register.
01	An interrupt is generated after every two bytes that are transmitted.
10	An interrupt occurs after every third byte that is transmitted.
11	An interrupt occurs after every fourth byte that is transmitted.

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

The transmit interrupt is cleared by a read of the status register. The status of this interrupt can be read by reading SPIxSTA[5]. The interrupt is disabled if SPIxCON[13] is left high.

A write to the control register, SPIxCON, resets the transmitted byte counter back to zero. For example, in a case where SPIxCON[15:14] is set to 0x3 and SPIxCON is written to after three bytes are transmitted, the Tx interrupt does not occur until another four bytes are transmitted.

Rx Interrupt

If TIM (SPIxCON[6]) is cleared, the Rx FIFO status causes the interrupt. SPIxCON[15:14] control when the interrupt occurs. The interrupt is cleared by a read of SPIxSTA. The status of this interrupt can be read by reading SPIxSTA[6].

Interrupts are only generated when data is written to the FIFO. For example, if SPIxCON[15:14] are set to 0x00, an interrupt is generated after the first byte is received. When the status register is read, the interrupt is deactivated. If the byte is not read from the FIFO, the interrupt is not regenerated. Another interrupt is not generated until another byte is received into the FIFO.

The interrupt depends on the number of valid bytes in FIFO and not the number of bytes received. For example, when SPIxCON[15:14] are set to 0x1, an interrupt is generated after a byte is received if there are two or more bytes in the FIFO. The interrupt is not generated after every two bytes received.

The interrupt is disabled if SPIxCON[12] is left high.

Underrun/Overflow Interrupts

SPIxSTA[7] and SPIxSTA[4] generate SPI interrupts.

When a transfer starts with no data in the Tx FIFO, SPIxSTA[4] is set to indicate an underrun condition. This causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of SPIxCON[15:14]. This interrupt is disabled if SPIxCON[13] is set.

When data is received and the Rx FIFO is already full, SPIxSTA[7] is set to 1, indicating an overflow condition. This causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of SPIxCON[15:14]. This interrupt is disabled if SPIxCON[12] is set.

When the SPI Rx overflow bit is set (SPIxSTA[7] set to 1), the contents of the SPI Rx FIFO is undetermined and must not be used. The user must flush the Rx FIFO on detecting this error condition.

All interrupts are cleared either by a read of the status register or if SPIxCON[0] is deasserted. The Rx and Tx interrupts are also cleared if the relevant flush bits are asserted. Otherwise, the interrupts remain active even if the SPI is reconfigured.

SPI WIRE-OR'ED MODE (WOM)

To prevent contention when the SPI is used in a multimaster or multislave system, the data output pins (MOSI and MISO) can be configured to behave as open-circuit drivers. An external pull-up resistor is required when this feature is selected. The WOM bit (SPIxCON[4]) controls the pad enable outputs for the data lines.

SPI CSERR CONDITION

The CSERR bit (SPIxSTA[12]) indicates if an erroneous deassertion of the $\overline{CS}$ signal has been detected before the completion of all eight SCLK cycles. This bit generates an interrupt and is available in all modes of operation: slave, master, and during DMA transfers.

If an interrupt occurs, generated by the CSERR bit (SPIxSTA[12]), the SPI enable bit (SPIxCON[0]) must be disabled and restarted to enable a clean recovery. This ensures that subsequent transfers are error free. The BCRST bit (SPIxDIV[7]) must be set at all times in both slave 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.

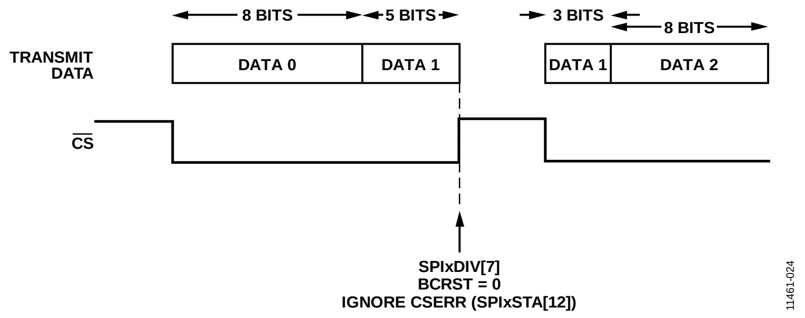


Figure 26. SPI Communication: Midbyte Stall

Note that the SPI must only be reenabled when the $\overline{CS}$ signal is high.

SPI DMA

DMA operation is provided on both SPI channels. Two DMA channels are dedicated to transmit and receive. The SPI DMA channels must be configured in the µDMA controller of the ARM Cortex-M3 processor.

It is possible to enable a DMA request on one or two channels at the same time by setting the DMA request bits for receive or transmit in the SPIxDMA register. If only the DMA transmit request (SPIxDMA[1]) is enabled, the Rx FIFO overflows during a SPI transfer, unless the received data is read by user code, and an overflow interrupt is generated. To avoid generating overflow interrupts, set the Rx FIFO flush bit, or disable the SPI interrupt in the NVIC. If only the DMA receive request (SPIxDMA[2]) is enabled, the Tx FIFO is underrun. To avoid an underrun interrupt, disable the SPI interrupt.

The SPI Tx (SPIxSTA[5]) and SPI Rx (SPIxSTA[6]) interrupts are not generated when using DMA. The SPI TXUR (SPIxSTA[4]) and RXOF (SPIxSTA[7]) interrupts are generated when using DMA. SPIxCON[15:14] are not used in transmit mode and must be set to 0x00 in receive mode.

The enable bit (SPIxDMA[0]) controls the start of a DMA transfer. DMA requests are only generated when the enable bit is set to 1. At the end of a DMA transfer, that is, when receiving a DMA SPI transfer interrupt, this bit must be cleared to prevent extra DMA requests to the µDMA controller. The data still present in the Tx FIFO is transmitted if in Tx mode.

DMA Master Transmit Configuration

The DMA SPI Tx channel must be configured.

The NVIC must be configured to enable DMA Tx master interrupt, for example, SPI0 Tx (ISER0[26]).

When all data present in the DMA buffer is transmitted, the DMA generates an interrupt. User code must disable the DMA request. Data is still in the Tx FIFO because the DMA request is generated each time there is free space in the Tx FIFO to always keep the FIFO full. User code can check how many bytes are still present in the FIFO in the FIFO status register.

DMA Master Receive Configuration

The SPIxCNT register is available in DMA receive master mode only. It sets the number of receive bytes required by the SPI master, or the number of clocks that the master must generate. When the required number of bytes are received, no more transfers are initiated. To initiate a DMA master receive transfer, user code must complete a dummy read. This dummy read must be added to the SPIxCNT number.

The counter counting the bytes as they are received is reset either when SPI is disabled in SPIxCON[0] or if the SPIxCNT register is modified by user code.

Performing SPIx DMA Master Receive

The DMA SPI Rx channel must be configured.

The NVIC must be configured to enable DMA Rx master interrupt (ISER0[29]).

The DMA transfer stops when the number of bytes have been transferred. Note that the DMA buffer must be of the same size as SPI1CNT to generate a DMA interrupt when the transfer is complete.

SPI AND POWER-DOWN MODES

In master mode, before entering power-down mode, it is recommended to disable the SPI block in SPIxCON[0]. In slave mode, in either mode of operation, interrupt driven or DMA, the $\overline{\text{CS}}$ line level must be checked via the GPIO registers to ensure that the SPI is not communicating and that the SPI block is disabled while the $\overline{\text{CS}}$ line is high. At power-up, the SPI block can be reenabled.

REGISTER SUMMARY: SPI0

Table 197. SPI0 Register Summary

Address	Name	Description	Reset	Access
0x4002C000	SPI0STA	Status register	0x0000	R
0x4002C004	SPI0RX	Receive register	0x0000	R
0x4002C008	SPI0TX	Transmit register	0x0000	W
0x4002C00C	SPI0DIV	Baud rate selection register	0x0000	RW
0x4002C010	SPI0CON	SPI configuration register	0x0000	RW
0x4002C014	SPI0DMA	SPI DMA enable register	0x0000	RW
0x4002C018	SPI0CNT	Transfer byte count register	0x0000	RW

REGISTER DETAILS: SPI0

Status Register

Address: 0x4002C000, Reset: 0x0000, Name: SPI0STA

Table 198. Bit Descriptions for SPI0STA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	CSRSG	Detected a rising edge on $\overline{CS}$, in CONT mode. This bit causes an interrupt. This can identify the end of an SPI data frame. 0: cleared to 0 when the status register is read. 1: set to 1 when there was a rising edge in the $\overline{CS}$ line, when the device was in master mode, continuous transfer, high frequency mode, and CSIRQ_EN was asserted.	0x0	RC
13	CSFLG	Detected a falling edge on $\overline{CS}$, in CONT mode. This bit causes an interrupt. This can identify the start of an SPI data frame. 0: cleared to 0 when the status register is read. 1: set to 1 when there was a falling edge in the $\overline{CS}$ line, when the device was in master mode, continuous transfer, high frequency mode, and CSIRQ_EN was asserted.	0x0	RC
12	CSERR	Detected a $\overline{CS}$ error condition. 0: cleared to 0 when the status register is read. 1: set to 1 when the $\overline{CS}$ line was deasserted abruptly, before the full byte of data was transmitted completely. This bit causes an interrupt.	0x0	RC
11	RXS	SPI Rx FIFO excess bytes present. 0: cleared to 0 when the number of bytes in the FIFO is less than or equal to the number indicated in the MOD bits (SPI0CON[15:14]). 1: set to 1 when the number of bytes in the Rx FIFO is greater than the number indicated in the MOD bits (SPI0CON[15:14]).	0x0	R
[10:8]	RXFSTA	SPI Rx FIFO status. 000: Rx FIFO empty. 001: 1 valid byte in the FIFO. 010: 2 valid bytes in the FIFO. 011: 3 valid bytes in the FIFO. 100: 4 valid bytes in the FIFO.	0x0	R
7	RXOF	SPI Rx FIFO overflow. 0: cleared when the SPISTA register is read. 1: set when the Rx FIFO was already full when new data was loaded to the FIFO. This bit generates an interrupt except when RFLUSH is set in SPI0CON.	0x0	RC
6	RX	SPI Rx IRQ. Not available in DMA mode. 0: cleared when the SPI0STA register is read. 1: set when a receive interrupt occurs. This bit is set when TIM in SPI0CON is cleared and the required number of bytes have been received.	0x0	RC

Bits	Bit Name	Description	Reset	Access
5	TX	SPI Tx IRQ status bit. Not available in DMA mode. 0: CLR. Cleared to 0 when the SPI0STA register is read. 1: SET. Set to 1 when a transmit interrupt occurs. This bit is set when TIM in SPI0CON is set and the required number of bytes have been transmitted.	0x0	RC
4	TXUR	SPI Tx FIFO underflow. 0: cleared to 0 when the SPI0STA register is read. 1: set to 1 when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt, except when TFLUSH is set in SPI0CON.	0x0	RC
[3:1]	TXFSTA	SPI Tx FIFO status. 000: Tx FIFO empty. 001: 1 valid byte in FIFO. 010: 2 valid bytes in FIFO. 011: 3 valid bytes in FIFO. 100: 4 valid bytes in FIFO.	0x0	R
0	IRQ	SPI interrupt status. 0: cleared to 0 after reading SPI0STA. 1: set to 1 when an SPI based interrupt occurs.	0x0	RC

Receive Register

Address: 0x4002C004, Reset: 0x0000, Name: SPI0RX

Table 199. Bit Descriptions for SPI0RX

Bits	Bit Name	Description	Reset	Access
[15:8]	DMA_DATA_BYTE_2	8-bit receive buffer. These 8 bits are used only in the DMA mode, where all FIFO accesses happen as half-word access. They return 0s if DMA is disabled.	0x0	R
[7:0]	DATA_BYTE_1	8-bit receive buffer.	0x0	R

Transmit Register

Address: 0x4002C008, Reset: 0x0000, Name: SPI0TX

Table 200. Bit Descriptions for SPI0TX

Bits	Bit Name	Description	Reset	Access
[15:8]	DMA_DATA_BYTE_2	8-bit transmit buffer. These 8 bits are used only in the DMA mode, where all FIFO accesses happen as half-word access. They return 0s if DMA is disabled.	0x0	W
[7:0]	DATA_BYTE_1	8-bit transmit buffer.	0x0	W

Baud Rate Selection Register

Address: 0x4002C00C, Reset: 0x0000, Name: SPI0DIV

Table 201. Bit Descriptions for SPI0DIV

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	CSIRQ_EN	Enable interrupt on every $\overline{\text{CS}}$ edge in CONT mode. If this bit is set and the SPI module is in continuous mode, any edge on $\overline{\text{CS}}$ generates an interrupt and the corresponding status bits (CSRSG, CSFLG) are asserted. If this bit is clear, no interrupt is generated. This bit has no effect if the SPI is not in continuous mode and high speed mode.	0x0	RW
7	BCRST	Reset Mode for CSERR. If this bit is set, the bit counter is reset after a $\overline{\text{CS}}$ error condition and the Cortex is expected to clear the SPI enable bit. If this bit is clear, the bit counter continues from where it stopped. SPI can receive the remaining bits when $\overline{\text{CS}}$ is asserted, and Cortex has to ignore the CSERR interrupt. However, it is strongly recommended to set this bit for a graceful recovery after a $\overline{\text{CS}}$ error.	0x0	RW

Bits	Bit Name	Description	Reset	Access
6	HFM	High frequency mode. This bit is used for applications using high frequency where the pad introduces a significant delay on the SCL. This can cause a significant enough difference between the serial clock and the data being received on the Rx shift register. In this mode, the Rx shift register is clocked by SCLIN instead of UCLK.	0x0	RW
[5:0]	DIV	SPI clock divider. DIV is the factor that divides HCLK to generate the serial clock.	0x0	RW

SPI Configuration Register

Address: 0x4002C010, Reset: 0x0000, Name: SPI0CON

Table 202. Bit Descriptions for SPI0CON

Bits	Bit Name	Description	Reset	Access
[15:14]	MOD	SPI IRQ mode bits. These bits configure when the Tx/Rx interrupts occur in a transfer. For DMA Rx transfer, these bits should be 00. 00: Tx interrupt occurs when 1 byte is transferred. Rx interrupt occurs when 1 or more bytes are received into the FIFO. 01: Tx interrupt occurs when 2 bytes are transferred. Rx interrupt occurs when 2 or more bytes are received into the FIFO. 10: Tx interrupt occurs when 3 bytes are transferred. Rx interrupt occurs when 3 or more bytes are received into the FIFO. 11: Tx interrupt occurs when 4 bytes are transferred. Rx interrupt occurs when the Rx FIFO is full, or 4 bytes present.	0x0	RW
13	TFLUSH	SPI Tx FIFO flush enable. 0: clear this bit to disable Tx FIFO flushing. 1: set this bit to flush the Tx FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is left high, either the last transmitted value or 0x00 is transmitted, depending on the ZEN bit. Any writes to the Tx FIFO are ignored while this bit is set.	0x0	RW
12	RFLUSH	SPI Rx FIFO flush enable. 0: clear this bit to disable Rx FIFO flushing. 1: set this bit to flush the Rx FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and TIM = 0, a read of the Rx FIFO initiates a transfer.	0x0	RW
11	CON	Continuous transfer enable. 0: DIS. Cleared by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPI0TX register, a new transfer is initiated after a stall period of 1 serial clock cycle. 1: EN. Set by user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. $\overline{\text{CS}}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty.	0x0	RW
10	LOOPBACK	Loopback enable. 0: cleared by user to be in normal mode. 1: set by user to connect MISO to MOSI and test software.	0x0	RW
9	OEN	Slave MISO output enable. 0: clear this bit to disable the output driver on the MISO pin. The MISO pin is open-circuit when this bit is clear. 1: set this bit for MISO to operate as normal.	0x0	RW
8	RXOF	SPIRX overflow overwrite enable. 0: cleared by user; the new serial byte received is discarded. 1: set by user; the valid data in the Rx register is overwritten by the new serial byte received.	0x0	RW
7	ZEN	Transmit zeros enable. 0: clear this bit to transmit the last transmitted value when there is no valid data in the Tx FIFO. 1: set this bit to transmit 0x00 when there is no valid data in the Tx FIFO.	0x0	RW

Bits	Bit Name	Description	Reset	Access
6	TIM	SPI transfer and interrupt mode. 0: cleared by user to initiate transfer with a read of the SPI0RX register. Interrupt only occurs when Rx is full. 1: set by user to initiate transfer with a write to the SPI0TX register. Interrupt only occurs when Tx is empty.	0x0	RW
5	LSB	LSB first transfer enable. 0: MSB transmitted first. 1: LSB transmitted first.	0x0	RW
4	WOM	SPI wired-OR mode. 1: open circuit data output enable. External pull-ups required on data out pins. 0: normal output levels.	0x0	RW
3	CPOL	Serial clock polarity. 0: serial clock idles low. 1: serial clock idles high.	0x0	RW
2	CPHA	Serial clock phase mode. 1: serial clock pulses at the beginning of each serial bit transfer. 0: serial clock pulses at the end of each serial bit transfer.	0x0	RW
1	MASEN	Master mode enable. 0: enable slave mode. 1: enable master mode.	0x0	RW
0	ENABLE	SPI enable. 0: disable the SPI. 1: enable the SPI.	0x0	RW

SPI DMA Enable Register

Address: 0x4002C014, Reset: 0x0000, Name: SPI0DMA

Table 203. Bit Descriptions for SPI0DMA

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
2	IENRXDMA	Enable receive DMA request. 0: disable Rx DMA interrupt. 1: enable Rx DMA interrupt.	0x0	RW
1	IENTXDMA	Enable transmit DMA request. 0: disable Tx DMA interrupt. 1: enable Tx DMA interrupt.	0x0	RW
0	ENABLE	Enable DMA for data transfer. Set by user code to start a DMA transfer. Cleared by user code at the end of DMA transfer. This bit must be cleared to prevent extra DMA requests to the μDMA controller.	0x0	RW

Transfer Byte Count Register

Address: 0x4002C018, Reset: 0x0000, Name: SPI0CNT

Table 204. Bit Descriptions for SPI0CNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	COUNT	Transfer byte count. COUNT indicates the number of bytes to be transferred. COUNT is used in both receive and transmit transfer types. The COUNT value assures that a master mode transfer terminates at the proper time and that 16-bit DMA transfers are byte padded or discarded as required to match odd transfer counts. Reset by clearing SPI0CON[0] or if SPI0CNT is updated.	0x0	RW

REGISTER SUMMARY: SPI1

Table 205. SPI1 Register Summary

Address	Name	Description	Reset	Access
0x40030000	SPI1STA	Status register	0x0000	R
0x40030004	SPI1RX	Receive register	0x0000	R
0x40030008	SPI1TX	Transmit register	0x0000	W
0x4003000C	SPI1DIV	Baud rate selection register	0x0000	RW
0x40030010	SPI1CON	SPI configuration register	0x0000	RW
0x40030014	SPI1DMA	SPI DMA enable register	0x0000	RW
0x40030018	SPI1CNT	Transfer byte count register	0x0000	RW

REGISTER DETAILS: SPI1

Status Register

Address: 0x40030000, Reset: 0x0000, Name: SPI1STA

Table 206. Bit Descriptions for SPI1STA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	CSRSG	Detected a rising edge on $\overline{\text{CS}}$, in CONT mode. This bit causes an interrupt. This can identify the end of an SPI data frame. 0: cleared to 0 when the status register is read. 1: set to 1 when there was a rising edge in $\overline{\text{CS}}$ line, when the device was in master mode, continuous transfer, high frequency mode, and CSIRQ_EN was asserted.	0x0	RC
13	CSFLG	Detected a falling edge on CS, in CONT mode. This bit causes an interrupt. This can identify the start of an SPI data frame. 0: cleared to 0 when the status register is read. 1: set to 1 when there was a falling edge in CS line, when the device was in master mode, continuous transfer, high frequency mode, and CSIRQ_EN was asserted.	0x0	RC
12	CSERR	Detected a $\overline{\text{CS}}$ error condition. 0: cleared to 0 when the status register is read. 1: set to 1 when the $\overline{\text{CS}}$ line was deasserted abruptly, even before the full byte of data was transmitted completely. This bit causes an interrupt.	0x0	RC
11	RXS	SPI Rx FIFO excess bytes present. 0: cleared to 0 when the number of bytes in the FIFO is less than or equal to the number indicated in the MOD bits (SPI0CON[15:14]). 1: set to 1 when the number of bytes in the Rx FIFO is greater than the number indicated in the MOD bits (SPI0CON[15:14]).	0x0	R
[10:8]	RXFSTA	SPI Rx FIFO status. 000: Rx FIFO empty. 001: 1 valid byte in the FIFO. 010: 2 valid bytes in the FIFO. 011: 3 valid bytes in the FIFO. 100: 4 valid bytes in the FIFO.	0x0	R
7	RXOF	SPI Rx FIFO overflow. 0: cleared to 0 when the SPI1STA register is read. 1: set to 1 when the Rx FIFO was already full when new data was loaded to the FIFO. This bit generates an interrupt, except when RFLUSH is set in SPI1CON.	0x0	RC
6	RX	SPI Rx IRQ. Not available in DMA mode. Set when a receive interrupt occurs. 0: cleared to 0 when the SPI1STA register is read. 1: set to 1 when TIM in SPI1CON is cleared and the required number of bytes have been received.	0x0	RC

Bits	Bit Name	Description	Reset	Access
5	TX	SPI Tx IRQ. Status bit. Not available in DMA mode. 0: CLR. Cleared to 0 when the SPI1STA register is read. 1: SET. Set to 1 when a transmit interrupt occurs. This bit is set when TIM in SPI1CON is set and the required number of bytes have been transmitted.	0x0	RC
4	TXUR	SPI Tx FIFO underflow. 0: cleared to 0 when the SPI1STA register is read. 1: set to 1 when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt, except when TFLUSH is set in SPI1CON.	0x0	RC
[3:1]	TXFSTA	SPI Tx FIFO status. 000: Tx FIFO empty. 001: 1 valid byte in FIFO. 010: 2 valid bytes in FIFO. 011: 3 valid bytes in FIFO. 100: 4 valid bytes in FIFO.	0x0	R
0	IRQ	SPI interrupt status. 0: cleared to 0 after reading SPI1STA. 1: set to 1 when an SPI based interrupt occurs.	0x0	RC

Receive Register

Address: 0x40030004, Reset: 0x0000, Name: SPI1RX

Table 207. Bit Descriptions for SPI1RX

Bits	Bit Name	Description	Reset	Access
[15:8]	DMA_DATA_BYTE_2	8-bit receive buffer. These 8 bits are used only in the DMA mode, where all FIFO accesses happen as half-word access. They return 0s if DMA is disabled.	0x0	R
[7:0]	DATA_BYTE_1	8-bit receive buffer.	0x0	R

Transmit Register

Address: 0x40030008, Reset: 0x0000, Name: SPI1TX

Table 208. Bit Descriptions for SPI1TX

Bits	Bit Name	Description	Reset	Access
[15:8]	DMA_DATA_BYTE_2	8-bit transmit buffer. These 8 bits are used only in the DMA mode, where all FIFO accesses happen as half-word access. They return 0s if DMA is disabled.	0x0	W
[7:0]	DATA_BYTE_1	8-bit transmit buffer.	0x0	W

Baud Rate Selection Register

Address: 0x4003000C, Reset: 0x0000, Name: SPI1DIV

Table 209. Bit Descriptions for SPI1DIV

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	CSIRQ_EN	Enable interrupt on every $\overline{CS}$ edge in continuous mode. If this bit is set and the SPI module is in continuous mode, any edge on $\overline{CS}$ generates an interrupt and the corresponding status bits (CSRSG, CSFLG) are asserted. If this bit is clear, no interrupt is generated. This bit has no effect if the SPI is not in continuous mode and high speed mode.	0x0	RW
7	BCRST	Reset mode for CSERR. If this bit is set, the bit counter is reset after a $\overline{CS}$ error condition and the Cortex is expected to clear the SPI enable bit. If this bit is clear, the bit counter continues from where it stopped. SPI can receive the remaining bits when $\overline{CS}$ is asserted, and Cortex must ignore the CSERR interrupt. However, it is strongly recommended to set this bit for a graceful recovery after a $\overline{CS}$ error.	0x0	RW

Bits	Bit Name	Description	Reset	Access
6	HFM	High frequency mode. This bit is used for applications using high frequency where the pad introduces a significant delay on the SCL, which can cause a significant enough difference between the serial clock and the data being received on the Rx shift register. In this mode, the Rx shift register is clocked by SCL instead of UCLK.	0x0	RW
[5:0]	DIV	SPI clock divider. DIV is the factor that divides HCLK to generate the serial clock.	0x0	RW

SPI Configuration Register

Address: 0x40030010, Reset: 0x0000, Name: SPI1CON

Table 210. Bit Descriptions for SPI1CON

Bits	Bit Name	Description	Reset	Access
[15:14]	MOD	SPI IRQ mode bits. These bits configure when the Tx/Rx interrupts occur in a transfer. For DMA Rx transfer, these bits must be 00. 00: Tx interrupt occurs when 1 byte is transferred. Rx interrupt occurs when 1 or more bytes are received into the FIFO. 01: Tx interrupt occurs when 2 bytes are transferred. Rx interrupt occurs when 2 or more bytes are received into the FIFO. 10: Tx interrupt occurs when 3 bytes are transferred. Rx interrupt occurs when 3 or more bytes are received into the FIFO. 11: Tx interrupt occurs when 4 bytes are transferred. Rx interrupt occurs when the Rx FIFO is full, or 4 bytes present.	0x0	RW
13	TFLUSH	SPI Tx FIFO flush enable. 0: clear this bit to disable Tx FIFO flushing. 1: set this bit to flush the Tx FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is left high, either the last transmitted value or 0x00 is transmitted depending on the ZEN bit. Any writes to the Tx FIFO are ignored while this bit is set.	0x0	RW
12	RFLUSH	SPI Rx FIFO flush enable. 0: clear this bit to disable Rx FIFO flushing. 1: set this bit to flush the Rx FIFO. This bit does not clear itself and must be toggled if a single flush is required. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and TIM = 0, a read of the Rx FIFO initiates a transfer.	0x0	RW
11	CON	Continuous transfer enable. 0: DIS. Cleared by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPI1TX register, a new transfer is initiated after a stall period of 1 serial clock cycle. 1: EN. Set by user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. $\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty.	0x0	RW
10	LOOPBACK	Loopback enable. 0: cleared by user to be in normal mode. 1: set by user to connect MISO to MOSI and test software.	0x0	RW
9	OEN	Slave MISO output enable. 0: clear this bit to disable the output driver on the MISO pin. The MISO pin is open-circuit when this bit is clear. 1: set this bit for MISO to operate as normal.	0x0	RW
8	RXOF	SPIRX overflow overwrite enable. 0: cleared by user; the new serial byte received is discarded. 1: set by user; the valid data in the Rx register is overwritten by the new serial byte received.	0x0	RW
7	ZEN	Transmit zeros enable. 0: clear this bit to transmit the last transmitted value when there is no valid data in the Tx FIFO. 1: set this bit to transmit 0x00 when there is no valid data in the Tx FIFO.	0x0	RW

Bits	Bit Name	Description	Reset	Access
6	TIM	SPI transfer and interrupt mode. 0: cleared by user to initiate transfer with a read of the SPIRX register. Interrupt only occurs when Rx is full. 1: set by user to initiate transfer with a write to the SPITX register. Interrupt only occurs when Tx is empty.	0x0	RW
5	LSB	LSB first transfer enable. 0: MSB transmitted first. 1: LSB transmitted first.	0x0	RW
4	WOM	SPI wired-OR mode. 0: normal output levels. 1: open circuit data output enable. External pull-ups required on data out pins.	0x0	RW
3	CPOL	Serial clock polarity. 0: serial clock idles low. 1: serial clock idles high.	0x0	RW
2	CPHA	Serial clock phase mode. 0: serial clock pulses at the end of each serial bit transfer. 1: serial clock pulses at the beginning of each serial bit transfer.	0x0	RW
1	MASEN	Master mode enable. 0: enable slave mode. 1: enable master mode.	0x0	RW
0	ENABLE	SPI enable. 0: disable the SPI. 1: enable the SPI.	0x0	RW

SPI DMA Enable Register

Address: 0x40030014, Reset: 0x0000, Name: SPI1DMA

Table 211. Bit Descriptions for SPI1DMA

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
2	IENRXDMA	Enable receive DMA request. 0: disable Rx DMA interrupt. 1: enable Rx DMA interrupt.	0x0	RW
1	IENTXDMA	Enable transmit DMA request. 0: disable Tx DMA interrupt. 1: enable Tx DMA interrupt.	0x0	RW
0	ENABLE	Enable DMA for data transfer. Set by user code to start a DMA transfer. Cleared by user code at the end of DMA transfer. This bit must be cleared to prevent extra DMA requests to the µDMA controller.	0x0	RW

Transfer Byte Count Register

Address: 0x40030018, Reset: 0x0000, Name: SPI1CNT

Table 212. Bit Descriptions for SPI1CNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	COUNT	Transfer byte count. COUNT indicates the number of bytes to be transferred. COUNT is used in both receive and transmit transfer types. The COUNT value assures that a master mode transfer terminates at the proper time and that 16-bit DMA transfers are byte padded or discarded as required to match odd transfer counts. Reset by clearing SPI1CON[0] or if SPI1CNT is updated.	0x0	RW

UART SERIAL INTERFACE

UART FEATURES

The ADuCM310 features an industry standard, 16450 UART peripheral with support for DMA.

UART OVERVIEW

The UART peripheral is a full-duplex universal asynchronous receiver/transmitter (UART), compatible with the industry standard, 16450. The UART is responsible for converting data between serial and parallel formats. The serial communication follows an asynchronous protocol, supporting various word length, stop bits, and parity generation options.

This UART also contains interrupt handling hardware. The UART features a fractional divider that facilitates high accuracy baud rate generation.

Interrupts can be generated from a number of unique events, such as full/empty data buffer, transfer error detection, and break detection.

UART OPERATION

Serial Communications

An asynchronous serial communication protocol is followed with these options:

- Five to eight data bits
- One, two, or 1½ stop bits
- None, even, or odd parity
- *Baud rate* = *UCLK/CDPCLK* ÷ (2 × 16 × *COMDIV*) ÷ (*M* + *N* ÷ 2048)
  where:
  UCLK/CDPCLK is the divided 80 MHz clock as configured via CLKCON1[10:8].
  COMDIV = 1 to 65536.
  M = 1 to 3.
  N = 0 to 2047.

All data-words require a start bit and at least one stop bit. This creates a range from 7 bits to 12 bits for each word. Transmit operation is initiated by writing to the transmit holding register (COMTX). After a synchronization delay, the data is moved to the internal transmit shift register (TSR), where it is shifted out at a baud (bit) rate equal to UCLK/CDPCLK ÷ (2 × 16 × COMDIV) ÷ (M + N ÷ 2048) with start, stop, and parity bits appended as required. All data-words begin with a low going start bit. The transfer of COMTX to the TSR causes the transmit register empty status flag to be set.

Receive operation uses the same data format as the transmit configuration except for the number of stop bits, which is always one. After detection of the start bit, the received word is shifted in the internal receive shift register (RSR). After the appropriate number of bits (including stop bits) are received, the data and any status is updated, and the RSR is transferred to the receive buffer register (COMRX). The receive buffer register full status flag is updated upon the transfer of the received word to this buffer and the appropriate synchronization delay.

A sampling clock equal to 16 times the baud rate samples the data as close to the midpoint of the bit as possible. A receive filter is also present that removes spurious pulses of less than two times the sampling clock period.

Note that data is transmitted and received least significant bit first. This is often not the assumed case by the user. However, it is standard for the protocol.

For power saving purposes, it is possible to disable the system clock to the UART via the CLKCON5[5] register. By default, the clock to the UART is disabled (CLKCON5[5] = 1).

Programmed Input/Output Mode

In programmed input/output mode, the software is responsible for moving data to and from the UART. This is typically accomplished by interrupt service routines that respond to the transmit and receive interrupts by either reading or writing data as appropriate. This mode places certain constraints on the software itself in that the software must respond within a certain time to prevent overflow errors from occurring in the receive channel.

Polling the status flag is processor intensive and not typically used unless the system can tolerate the overhead. Interrupts can be disabled using the COMIEN register.

Avoid writing to the COMTX register when it is not empty or reading the COMRX when it is not full, because this produces incorrect results. In the former case, COMTX is overwritten by the new word, and the previous word is never transmitted. In the latter case, the previously received word is read again. Both of these errors must be avoided in software by correctly using either interrupts or the status register polling. These errors are not detected in the hardware.

Enable/Disable Bit

Before the ADuCM310 enters power-down mode, it is recommended to disable the serial interfaces. A bit is provided in the UART control register to disable the UART serial peripheral. This bit disables the clock to the peripheral. When setting this bit, take care in the software that no data is being transmitted or received. If set during communication, the data transfer does not complete; the receive or transmit register contains only part of the data.

Interrupts

The UART peripheral has one interrupt output to the interrupt controller for both Rx and Tx interrupts. The COMIIR register must be read by software to determine the cause of the interrupt. Note that in DMA mode, the break interrupt is not available.

In input/output mode, when receiving, the interrupt is generated for the following cases:

- COMRX full
- Receive overflow error
- Receive parity error
- Receive framing error
- Break interrupt (UART RxD held low)
- COMTX empty

Buffer Requirements

This UART is double buffered (holding register and shift register).

DMA Mode

In DMA mode, user code does not move data to and from the UART. DMA request signals going to the external DMA block indicate that the UART is ready to transmit or receive data. These DMA request signals can be disabled in the COMIEN register.

Example Code to Set Up UART Receive DMA Channel

```
void UARTRXDMAINIT(void)
{
NVIC_EnableIRQ(DMA_UART_RX_IRQn);                          // UArt Tx DMA interrupt enable
pADI_UART->COMLCR = COMLCR_WLS_EIGHTBITS | COMLCR_STOP;     // 8-data bits + 1 Stop bit.
pADI_UART->COMDIV = 0x41;                                   // Set UART Baud rate
pADI_UART->COMFBR = COMFBR_FBEN_EN | 0x803;                 // DIVM = 1, DIVN = 3.
pADI_GP1->GPCON = 0x5;                                      // Configure P1.0/P1.1 for UART
Dma_Init();
pADI_DMA->DMACFG = 0x1;
UARTDMAREAD(uxUARTRXData, 4);
                                              // Enable DMA mode in DMA controller
pADI_DMA->DMAENSET = 0x20;                                  // Enable UART_RX_DMA Channel
pADI_UART->COMIEN = 0x20;                                   // Enable DMA Rx transfers
}

void UARTDMAREAD(unsigned char *pucRX_DMA, unsigned int iNumVals)
{
   DmaDesc Desc;
   // Common configuration of all the descriptors used here
   Desc.ctrlCfg.bits.cycle_ctrl       = DMA_BASIC;
   desc.ctrlcfg.bits.next_useburst    = 0x0;
   desc.ctrlcfg.bits.r_power          = 0;
   Desc.ctrlCfg.Bits.src_prot_ctrl    = 0x0;
   Desc.ctrlCfg.Bits.dst_prot_ctrl    = 0x0;
   Desc.ctrlCfg.Bits.src_size         = DMA_SIZE_BYTE;
   Desc.ctrlCfg.Bits.dst_size         = DMA_SIZE_BYTE;
   //  RX Primary   Descriptor
   Desc.srcEndPtr                     = (unsigned int)(&pADI_UART->COMRX);
   Desc.destEndPtr                    = (unsigned int)(pucTX_DMA + (iNumVals - 0x1) );
   Desc.ctrlCfg.Bits.n_minus_1        = iNumRX - 0x1;
   Desc.ctrlCfg.Bits.src_inc          = DMA_SRCINC_NO;
   Desc.ctrlCfg.Bits.dst_inc          = DMA_DSTINC_BYTE;
   *Dma_GetDescriptor(UARTRX_C) = Desc;
}
                                              // UART DMA Rx IRQ handler
void DMA_UART_RX_Int_Handler()
{
   NVIC_DisableIRQ(DMA_UART_RX_IRQn);                       // Clear Interrupt source
}
```

Example Code to Set Up UART Transmit DMA Channel

```
void UARTTXDMAINIT(void)
{
  NVIC_EnableIRQ(DMA_UART_TX_IRQn);                           // UART Tx DMA interrupt sources
  pADI_UART->COMLCR = COMLCR_WLS_8BITS + COMLCR_STOP;         // 8-data bits + 1 Stop bit.
  pADI_UART->COMDIV = 0x41;                                   // Set UART Baud rate
  pADI_UART->COMFBR = COMFBR_FBEN_EN | 0x803;                 // DIVM = 1, DIVN = 3.
  pADI_GP1->GPCON = 0x5;                                      // Configure P1.0/P1.1 for UART
  Dma_Init();
  pADI_DMA->DMACFG = 0x1;                                     // Enable DMA mode in DMA controller
  UARTDMAWRITE(uxUARTTXData, 16);
  pADI_DMA->DMAENSET = 0x10;                                  // Enable UART_TX_DMA Channel
  pADI_UART->COMIEN = 0x10;                                   // Enable DMA Tx transfers
}



void UARTDMAWRITE(unsigned char *pucTX_DMA, unsigned int iNumVals)
{
   DmaDesc Desc;

    // Common configuration of all the descriptors used here
    Desc.ctrlCfg.Bits.cycle_ctrl       = DMA_BASIC;
    Desc.ctrlCfg.Bits.next_useburst    = 0x0;
    Desc.ctrlCfg.Bits.r_power          = 0;
    Desc.ctrlCfg.Bits.src_prot_ctrl    = 0x0;
    Desc.ctrlCfg.Bits.dst_prot_ctrl    = 0x0;
    Desc.ctrlCfg.Bits.src_size         = DMA_SIZE_BYTE;
    Desc.ctrlCfg.Bits.dst_size         = DMA_SIZE_BYTE;

                                              //  TX Primary   Descriptor

    Desc.srcEndPtr                     = (unsigned int)(pucTX_DMA + (iNumVals - 0x1) );
    Desc.destEndPtr                    = (unsigned int)(&pADI_UART->COMTX);
    Desc.ctrlCfg.Bits.n_minus_1        = iNumRX - 0x1;
    Desc.ctrlCfg.Bits.src_inc          = DMA_SRCINC_BYTE;
    Desc.ctrlCfg.Bits.dst_inc          = DMA_DSTINC_NO;
    *Dma_GetDescriptor(UARTTX_C) = Desc;

}

                                              // UART DMA Tx IRQ handler
void DMA_UART_TX_Int_Handler()
{
   NVIC_DisableIRQ(DMA_UART_TX_IRQn);                         // Clear Interrupt source
}
```

REGISTER SUMMARY: UART

Table 213. UART Register Summary

Address	Name	Description	Reset	Access
0x40005000	COMTX	Transmit holding register	0x0000	W
0x40005000	COMRX	Receive buffer register	0x0000	R
0x40005004	COMIEN	Interrupt enable register	0x0000	RW
0x40005008	COMIIR	Interrupt identification register	0x0001	RC
0x4000500C	COMLCR	Line control register	0x0000	RW
0x40005010	COMMCR	Modem control register	0x0000	RW
0x40005014	COMLSR	Line status register	0x0060	RC
0x40005018	COMMSR	Modem status register	0x0000	RC
0x4000501C	COMSCR	Scratch buffer register	0x0000	RW
0x40005024	COMFBR	Fractional baud rate register	0x0000	RW
0x40005028	COMDIV	Baud rate divider register	0x0001	RW

REGISTER DETAILS: UART

Transmit Holding Register

Address: 0x40005000, Reset: 0x0000, Name: COMTX

COMRX and COMTX share the same address while they are implemented as different registers. If these registers are written to, the user accesses the transmit holding register (COMTX). If these registers are read from, the user accesses the receive buffer register (COMRX).

Table 214. Bit Descriptions for COMTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	THR	Transmit holding register. This is an 8-bit register to which the user can write the data to be sent. If the ETBEI bit is set in the COMIEN register, an interrupt is generated when COMTX is empty. If user code sets ETBEI while COMTX is already empty, an interrupt is generated immediately.	0x0	W

Receive Buffer Register

Address: 0x40005000, Reset: 0x0000, Name: COMRX

Table 215. Bit Descriptions for COMRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	RBR	Receive buffer register. This is an 8-bit register from which the user can read received data. If the ERBFI bit is set in the COMIEN register, an interrupt is generated when this register is fully loaded with the received data via the serial input port. If user code sets the ERBFI bit while COMRX is already full, an interrupt is generated immediately.	0x0	R

Interrupt Enable Register

Address: 0x40005004, Reset: 0x0000, Name: COMIEN

COMIEN is the interrupt enable register that configures which interrupt source generates the interrupt. Only the lowest four bits in this register enable interrupts. Bit 4 and Bit 5 enable UART DMA signals. The UART DMA channel and interrupt must be configured in the DMA block.

Table 216. Bit Descriptions for COMIEN

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved.	0x0	R
5	EDMAR	DMA requests in receive mode. 0: DMA requests disabled. 1: DMA requests enabled.	0x0	RW
4	EDMAT	DMA requests in transmit mode. 0: DMA requests disabled. 1: DMA requests enabled.	0x0	RW
3	EDSSI	Modem status interrupt. Interrupt is generated when any of COMMSR[3:0] are set. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
2	ELSI	Rx status interrupt. 0: interrupt disabled. 1: interrupt enabled	0x0	RW
1	ETBEI	Transmit buffer empty interrupt. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW
0	ERBFI	Receive buffer full interrupt. 0: interrupt disabled. 1: interrupt enabled.	0x0	RW

Interrupt Identification Register

Address: 0x40005008, Reset: 0x0001, Name: COMIIR

Table 217. Bit Descriptions for COMIIR

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0	R
[2:1]	STA	Interrupt status. When NIRQ is low (active low), this indicates an interrupt, and the following STA bit decoding is used. 00: modem status interrupt (read COMMSR to clear). 01: transmit buffer empty interrupt (write to COMTX or read COMIIR to clear). 10: receive buffer full interrupt (read COMRX to clear). 11: receive line status interrupt (read COMLSR to clear).	0x0	RC
0	NIRQ	Interrupt flag. 0: interrupt occurred. Source of interrupt indicated in the STA bits. 1: no interrupt occurred.	0x1	RC

Line Control Register

Address: 0x4000500C, Reset: 0x0000, Name: COMLCR

Table 218. Bit Descriptions for COMLCR

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	R
6	BRK	Set break. 0: force TxD to 0. 1: normal TxD operation	0x0	RW

Bits	Bit Name	Description	Reset	Access
5	SP	Stick parity. Forces parity to defined values. When set, the parity is based on the following bit settings: EPS = 1 and PEN = 1, parity is forced to 0. EPS = 0 and PEN = 1, parity is forced to 1. EPS = X and PEN = 0, no parity is transmitted. 0: parity is not forced based on EPS and PEN. 1: parity forced based on EPS and PEN.	0x0	RW
4	EPS	Parity select. This bit only has meaning if parity is enabled (PEN set). 0: odd parity is transmitted and checked. 1: even parity is transmitted and checked.	0x0	RW
3	PEN	Parity enable. Controls the parity bit transmitted and checked. The value transmitted and the value checked are based on the settings of EPS and SP. 0: parity is not transmitted or checked. 1: parity is transmitted and checked.	0x0	RW
2	STOP	Stop bit. Controls the number of stop bits transmitted. In all cases, only the first stop bit is evaluated on data received. 0: send 1 stop bit regardless of the word length (WLS). 1: send a number of stop bits based on the word length. Transmit 1.5 stop bits if the word length is 5 bits (WLS = 00), or 2 stop bits if the word length is 6 (WLS = 01), 7 (WLS = 10), or 8 bits (WLS = 11).	0x0	RW
[1:0]	WLS	Word length select. Selects the number of bits per transmission. 00: 5 bits. 01: 6 bits. 10: 7 bits. 11: 8 bits.	0x0	RW

Modem Control Register

Address: 0x40005010, Reset: 0x0000, Name: COMMCR

Table 219. Bit Descriptions for COMMCR

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	R
4	LOOPBACK	Loopback mode. In loopback mode, SOUT is forced high. The modem signals are also directly connected to the status inputs (RTS to CTS, DTR to DSR, OUT1 to RI, and OUT2 to DCD). 0: normal operation; loopback disabled. 1: loopback enabled.	0x0	RW
3	OUT2	Output 2. 0: force OUT2 to a Logic 1. 1: force OUT2 to a Logic 0.	0x0	RW
2	OUT1	Output 1. 0: force OUT1 to a Logic 1. 1: force OUT1 to a Logic 0.	0x0	RW
1	RTS	Request to send. 0: force RTS to a Logic 1. 1: force RTS to a Logic 0.	0x0	RW
0	DTR	Data terminal ready. 0: force DTR to a Logic 1. 1: force DTR to a Logic 0.	0x0	RW

Line Status Register

Address: 0x40005014, Reset: 0x0060, Name: COMLSR

Table 220. Bit Descriptions for COMLSR

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x0	R
6	TEMT	COMTX and shift register empty status. 0: COMTX has been written to and contains data to be transmitted. Take care not to overwrite its value. 1: COMTX and the transmit shift register are empty, and it is safe to write new data to COMTX. Data has been transmitted.	0x1	R
5	THRE	COMTX empty. THRE is cleared when COMRX is read. 0: COMTX has been written to and contains data to be transmitted. Take care taken not to overwrite its value. 1: COMTX is empty, and it is safe to write new data to COMTX. The previous data may not have been transmitted yet and may still be present in the shift register.	0x1	R
4	BI	Break indicator. If set, this bit self clears after COMLSR is read. 0: SIN is not detected to be longer than the maximum word length. 1: SIN is held low for more than the maximum word length.	0x0	RC
3	FE	Framing error. If set, this bit self clears after COMLSR is read. 0: no invalid stop bit is detected. 1: an invalid stop bit is detected on a received word.	0x0	RC
2	PE	Parity error. If set, this bit self clears after COMLSR is read. 0: no parity error is detected. 1: a parity error has occurred on a received word.	0x0	RC
1	OE	Overrun error. If set, this bit self clears after COMLSR is read. 0: receive data has not been overwritten. 1: receive data was overwritten by new data before COMRX was read.	0x0	RC
0	DR	Data ready. This bit is cleared only by reading COMRX. It does not self clear. 0: COMRX does not contain new receive data. 1: COMRX contains receive data to be read.	0x0	RC

Modem Status Register

Address: 0x40005018, Reset: 0x0000, Name: COMMSR

Table 221. Bit Descriptions for COMMSR

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	DCD	Data carrier detect. This bit reflects the direct status complement of the DCD pin. 0: DCD is logic high. 1: DCD is logic low.	0x0	R
6	RI	Ring indicator. This bit reflects the direct status complement of the DCD pin. 0: RI is logic high. 1: RI is logic low.	0x0	R
5	DSR	Data set ready. This bit reflects the direct status complement of the DCD pin. 0: DSR is logic high. 1: DSR is logic low.	0x0	R
4	CTS	Clear to send. This bit reflects the direct status complement of the DCD pin. 0: CTS is logic high. 1: CTS is logic low.	0x0	R
3	DDCD	Delta DCD. If set, this bit self clears after COMMSR is read. 0: DCD has not changed state since COMMSR was last read. 1: DCD has changed state since COMMSR last read.	0x0	R

Bits	Bit Name	Description	Reset	Access
2	TERI	Trailing edge RI. If set, this bit self clears after COMMSR is read. 0: RI has not changed from 0 to 1 since COMMSR was last read. 1: RI changed from 0 to 1 since COMMSR was last read.	0x0	R
1	DDSR	Delta DSR. If set, this bit self clears after COMMSR is read. 0: DSR has not changed state since COMMSR was last read. 1: DSR changed state since COMMSR was last read.	0x0	R
0	DCTS	Delta CTS. If set, this bit self clears after COMMSR is read. 0: CTS has not changed state since COMMSR was last read. 1: CTS changed state since COMMSR was last read.	0x0	R

Scratch Buffer Register

Address: 0x4000501C, Reset: 0x0000, Name: COMSCR

Table 222. Bit Descriptions for COMSCR

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
[7:0]	SCR	Scratch. The scratch register is an 8-bit register stores intermediate results. The value contained in the scratch register does not affect UART functionality or performance. Only 8 bits of this register are implemented. Bits[15:8] are read only and always return 0x00 when read. SCR is writable with any value from 0 to 255. A read returns the last value written.	0x0	RW

Fractional Baud Rate Register

Address: 0x40005024, Reset: 0x0000, Name: COMFBR

Table 223. Bit Descriptions for COMFBR

Bits	Bit Name	Description	Reset	Access
15	FBEN	Fractional baud rate generator enable. The generating of fractional baud rate can be described by the following formula and the final baud rate of UART operation is calculated as $Baud\ Rate = (UCLK/CDPCLK/(2 \times (M + N/2048))\ 16 \times COMDIV$	0x0	RW
[14:13]	RESERVED	Reserved.	0x0	R
[12:11]	DIVM	Fractional baud rate M divide Bit 1 to Bit 3. These bits must not be 0.	0x0	RW
[10:0]	DIVN	Fractional baud rate N divide Bit 0 to Bit 2047.	0x0	RW

Baud Rate Divider Register

Address: 0x40005028, Reset: 0x0001, Name: COMDIV

Table 224. Bit Descriptions for COMDIV

Bits	Bit Name	Description	Reset	Access
[15:0]	DIV	Baud rate divider. The COMDIV register must not be 0, which is not specified. The range of allowed DIV values is from 1 to 65,535.	0x1	RW

GENERAL-PURPOSE TIMERS

GENERAL-PURPOSE TIMERS FEATURES

The ADuCM310 integrates three general-purpose timers with the following features:

- Three identical, general-purpose, 16-bit count-up/count-down timers
    - Timer 0, Timer 1, and Timer 2
- Clocked from four different clocks
    - Peripheral clock (PCLK)
    - 80 MHz system clock (HCLK)
    - 32 kHz internal oscillator (LFOSC)
    - 16 MHz external crystal (HFXTAL) or internal 16 MHz oscillator (HFOSC), dependent on CLKCON0[11]
- Clock sources can be scaled down using a prescaler of 16,256, or 32,768. Additionally, two of the clocks can be scaled down using a prescaler of 4, while the other two clock sources can be used directly (prescaler of 1).
- Two modes
    - Free running
    - Periodic
- Capture events feature
    - Capability to capture 15 different events on each timer

GENERAL-PURPOSE TIMERS BLOCK DIAGRAM

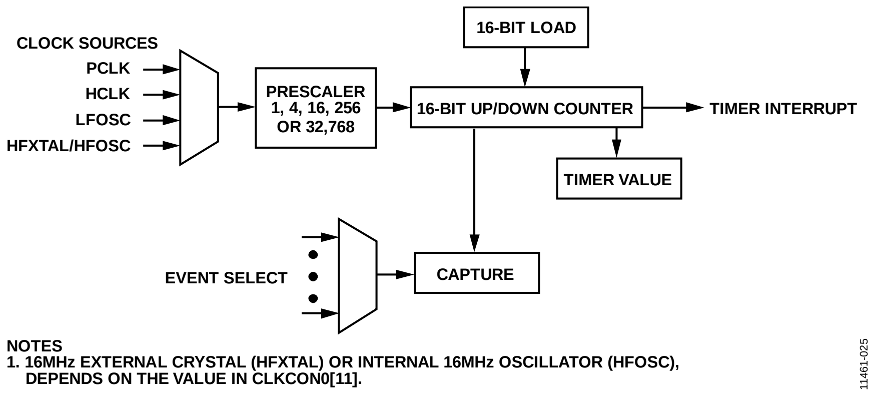

CLOCK SOURCES
PCLK
HCLK
LFOSC
HFXTAL/HFOSC
PRESCALER 1, 4, 16, 256 OR 32,768
16-BIT LOAD
16-BIT UP/DOWN COUNTER
TIMER INTERRUPT
TIMER VALUE
EVENT SELECT
CAPTURE
NOTES
1. 16MHz EXTERNAL CRYSTAL (HFXTAL) OR INTERNAL 16MHz OSCILLATOR (HFOSC), DEPENDS ON THE VALUE IN CLKCON0[11].
11461-025

Figure 27. General-Purpose Timers Block Diagram

GENERAL-PURPOSE TIMERS OVERVIEW

Timer 0, Timer 1, and Timer 2 are three identical, general-purpose, 16-bit count-up/count-down timers. They can be clocked from four different clock sources:

- PCLK
- HCLK
- 32 kHz internal oscillator (LFOSC)
- 16 MHz external crystal (HFXTAL) or the internal 16 MHz oscillator (HFOSC); configured via CLKCON0[11]

This clock source can be scaled down using a prescaler of 1 or 4, 16,256, or 32,768.

The timers can be either free running or periodic.

- In free running mode, the counter decrements from full scale to zero scale or increments from zero scale to full scale, and then restarts.
- In periodic mode, the counter decrements or increments from the value in the load register (TxLD MMR, where x is 0 for Timer 0, 1 for Timer 1, and 2 for Timer 2) until zero scale or full scale is reached, and then restarts at the value stored in the load register.

The value of a counter can be read at any time by accessing its value register (TxVAL).

The TxCON register selects the timer mode, configures the clock source, selects count-up/count-down, starts the counter, and controls the event capture function.

An interrupt signal is generated each time the value of the counter reaches zero when counting down, or each time the counter value reaches the maximum value when counting up. An IRQ can be cleared by writing 1 to the time clear interrupt register of that particular timer (TxCLRI).

In addition, Timer 0, Timer 1, and Timer 2 have a capture register (TxCAP) that is triggered by a selected IRQ source initial assertion. When triggered, the current timer value is copied to TxCAP, and the timer continues to run. This feature can determine the assertion of an event with increased accuracy.

GENERAL-PURPOSE TIMERS OPERATION

Free Running Mode

In free running mode, the timer is started by setting the enable bit (TxCON[4]) to 1 and the MOD bit (TxCON[3]) to 0. The timer increments from zero scale/full scale to full scale/zero scale if counting up/down. Full scale is $2^{16} - 1$ or 0xFFFF in binary format. Upon reaching full scale (or zero scale), a timeout interrupt occurs, and TxSTA[0] is set. To clear the timer interrupt, user code must write 1 to TxCLRI[0]. If TxCON[7] is set, the timer keeps counting and reloads when the TxCLRI register is written.

Periodic Mode

In periodic mode, the initial TxLD value must be loaded before starting the timer by setting the enable bit (TxCON[4]) to 1. The timer value either increments from the value in TxLD to full scale or decrements from the value in TxLD to zero scale, depending on the TxCON[2] settings (count up/down). Upon reaching full scale or zero scale, the timer generates an interrupt. The TxLD is reloaded into TxVAL, and the timer continues counting up or down. The timer must be disabled prior to changing the TxCON or TxLD register. If the TxLD register is changed while the timer is being loaded, undefined results can occur. By default, the counter is reloaded automatically when generating the interrupt signal. If TxCON[7] is set to 1, the counter is also reloaded when user code writes TxCLRI. This allows user changes to the TxLD to take effect immediately and not on the next timeout.

The timer interval is calculated as follows.

If the timer is set to count down,

$$Interval = (TxLD \times Prescaler)/Source\ Clock$$

For example, if TxLD = 0x100, prescaler = 4, and clock source = UCLK, the interval is 12.8 μs (where UCLK = 80 MHz).

If the timer is set to count up,

$$Interval = ((Full\ Scale - TxLD) \times Prescaler)/Source\ Clock$$

Asynchronous Clock Source

Timers are started by setting the enable bit (TxCON[4]) to 1 in the control register of the corresponding timer.

However, when the timer clock source is HFXTAL or LFOSC, some precautions must be taken:

- The control register (TxCON) must not be written if TxSTA[6] is set. Therefore, TxSTA must be read prior to configuring the control register (TxCON). When TxSTA[6] is cleared, the register can be modified. This ensures that synchronizing the timer control between the processor and timer clock domains is complete. TxSTA[6] is the timer busy status bit.
- After clearing the interrupt in TxCLRI, ensure that the register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary and check that TxSTA[7] = 0.

```
__asm void asmDSB()
{
nop
DSB
BX LR
}
```

- The value of a counter can be read at any time by accessing its value register (TxVAL). In an asynchronous configuration, TxVAL must always be read twice. If the two readings are different, read it a third time to obtain the correct value.

TxSTA must be read prior to writing to any timer registers after setting or clearing the enable bit. When TxSTA[7] is cleared, registers can be modified. This ensures that the timer has completed synchronization between the processor and timer clock domains. The typical synchronization time is two timer clock periods.

The TxCON register enables the counter, selects the mode, selects the prescaler value, and controls the event capture function.

Capture Event Function

There are a number of interrupt events that can be captured by the general-purpose timers. These events are shown in Table 225. Any of the events associated with a general-purpose timer can cause a capture of the 16-bit TxVAL register into the 16-bit TxCAP register. TxCON has a 4-bit field selecting which of the events to capture.

When the selected interrupt event occurs, the TxVAL register is copied into the TxCAP register. TxSTA[1] is set, indicating that a capture event is pending. The bit is cleared by writing 1 to TxCLRI[1]. The TxCAP register also holds its value and cannot be overwritten until a 1 is written to TxCLRI[1].

Table 225. Capture Event Function

Event Select Bits (EVENT Bits in TxCON Register, TxCON[11:8])	Timer 0 Capture Source	Timer 1 Capture Source	Timer 2 Capture Source
0000	Wake-Up Timer	External Interrupt 4	External Interrupt 7
0001	External Interrupt 0	External Interrupt 5	External Interrupt 8
0010	External Interrupt 1	External Interrupt 6	SPI1
0011	External Interrupt 2	Flash controller	I^2C0 slave
0100	External Interrupt 3	UART	I^2C0 master
0101	External Interrupt 4	SPI0	PLA 2
0110	External Interrupt 5	PLA 0	PLA 3
0111	External Interrupt 6	PLA 1	PWM trip
1000	External Interrupt 7	DMA Error	PWM 0
1001	External Interrupt 8	DMA Done (Any)	PWM 1
1010	Watchdog timer	Reserved	PWM 2
1011	Reserved	Reserved	PWM 3
1100	Reserved	Reserved	Low Voltage Analog Die Interrupt 1
1101	Low Voltage Analog Die Interrupt 0	I^2C1 slave	External Interrupt 0
1110	Reserved	I^2C1 master	External Interrupt 1
1111	General-Purpose Timer 1	General-Purpose Timer 2	General-Purpose Timer 1

REGISTER SUMMARY: GENERAL-PURPOSE TIMER 0

Table 226. Timer 0 Register Summary

Address	Name	Description	Reset	Access
0x40000000	T0LD	16-bit load value register	0x0000	RW
0x40000004	T0VAL	16-bit timer value register	0x0000	R
0x40000008	T0CON	Control register	0x000A	RW
0x4000000C	T0CLRI	Clear interrupt register	0x0000	W
0x40000010	T0CAP	Capture register	0x0000	R
0x4000001C	T0STA	Status register	0x0000	R

REGISTER DETAILS: GENERAL-PURPOSE TIMER 0

16-Bit Load Value Register

Address: 0x40000000, Reset: 0x0000, Name: T0LD

Table 227. Bit Descriptions for T0LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value. The up/down counter is periodically loaded with this value if periodic mode is selected (T0CON[3] = 1). LOAD writes during up/down counter timeout events are delayed until the event has passed.	0x0	RW

16-Bit Timer Value Register

Address: 0x40000004, Reset: 0x0000, Name: T0VAL

Table 228. Bit Descriptions for T0VAL

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

Control Register

Address: 0x40000008, Reset: 0x000A, Name: T0CON

Table 229. Bit Descriptions for T0CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	EVENTEN	Event select. This bit enables and disables the capture of events. Used in conjunction with the EVENT select range: when a selected event occurs, the current value of the up/down counter is captured in T0CAP. 0: events are not captured. 1: events are captured.	0x0	RW
[11:8]	EVENT	Event select range. Timer event select range (0 to 15).	0x0	RW
7	RLD	Reload control. RLD is only used for periodic mode; this bit allows the user to select whether the up/down is reset only on a timeout event or also when T0CLRI[0] is set. 1: up/down counter is reset when T0CLRI[0] is set. 0: up/down counter is only reset on a timeout event.	0x0	RW
[6:5]	CLK	Clock select. These bits select a timer clock from the four available clock sources. 00: PCLK. 01: HCLK. 10: LFOSC (32 kHz oscillator). 11: HFXTAL, if CLKCON0[11] = 1. 11: HFOSC, if CLKCON0[11] = 0.	0x0	RW

Bits	Bit Name	Description	Reset	Access
4	ENABLE	Timer enable. This bit enables and disables the timer. Clearing this bit resets the timer, including the T0VAL register. 0: DIS. Timer is disabled (default). 1: EN. Timer is enabled.	0x0	RW
3	MOD	Timer mode. This bit controls whether the timer runs in periodic or free running mode. In periodic mode, the up/down counter starts at the defined LOAD value (T0LD); in free running mode, the up/down counter starts at 0x0000 or 0xFFFF depending on whether the timer is counting up or down. 0: FREERUN. Timer runs in free running mode. 1: PERIODIC. Timer runs in periodic mode (default).	0x1	RW
2	UP	Count up. This bit controls whether the timer increments (counts up) or decrements (counts down) the up/down counter. 0: DIS. Timer is set to count down (default). 1: EN. Timer is set to count up.	0x0	RW
[1:0]	PRE	Prescaler. These bits control the prescaler division factor applied to the selected clock of the timer. If CLK Source 0 or CLK Source 1 are selected, a prescaler value of 0 means divide by 4; otherwise, it means divide by 1. 00: Source Clock/[1 or 4]. 01: Source Clock/16. 10: Source Clock/256. 11: Source Clock/32,768.	0x2	RW

Clear Interrupt Register

Address: 0x4000000C, Reset: 0x0000, Name: T0CLRI

Table 230. Bit Descriptions for T0CLRI

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

Capture Register

Address: 0x40000010, Reset: 0x0000, Name: T0CAP

Table 231. Bit Descriptions for T0CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	CAP	16-bit captured value. T0CAP holds its value until T0CLRI[1] is set by user code. T0CAP is not overwritten even if another event occurs without writing to T0CLRI[1].	0x0	R

Status Register

Address: 0x4000001C, Reset: 0x0000, Name: T0STA

Table 232. Bit Descriptions for T0STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	PDOK	T0CLRI synchronization. This bit is set automatically when the user sets T0CLRI[0] = 1. It is cleared automatically when the clear interrupt request crosses clock domains and takes effect in the timer clock domain. 0: CLR. The interrupt is cleared in the timer clock domain. 1: SET. T0CLRI[0] is being updated in the timer clock domain.	0x0	R
6	BUSY	Timer busy. This bit informs the user that a write to T0CON is still crossing into the timer clock domain. Check this bit after writing T0CON, and suppress further writes until this bit is cleared. 0: CLR. Timer ready to receive commands to T0CON. 1: SET. Timer not ready to receive commands to T0CON.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: CLR. No capture event is pending. 1: SET. A capture event is pending.	0x0	R
0	TMOUT	Timeout event occurred. This bit is set automatically when the value of the counter reaches zero while counting down or reaches full scale when counting up. This bit is cleared when T0CLRI[0] is set by the user. 0: CLR. No timeout event has occurred. 1: SET. A timeout event has occurred.	0x0	R

REGISTER SUMMARY: GENERAL-PURPOSE TIMER 1

Table 233. Timer 1 Register Summary

Address	Name	Description	Reset	Access
0x40000400	T1LD	16-bit load value register	0x0000	RW
0x40000404	T1VAL	16-bit timer value register	0x0000	R
0x40000408	T1CON	Control register	0x000A	RW
0x4000040C	T1CLRI	Clear interrupt register	0x0000	W
0x40000410	T1CAP	Capture register	0x0000	R
0x4000041C	T1STA	Status register	0x0000	R

REGISTER DETAILS: GENERAL-PURPOSE TIMER 1

16-Bit Load Value Register

Address: 0x40000400, Reset: 0x0000, Name: T1LD

Table 234. Bit Descriptions for T1LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value. The up/down counter is periodically loaded with this value if periodic mode is selected (T1CON[3] = 1). LOAD writes during up/down counter timeout events are delayed until the event has passed.	0x0	RW

16-Bit Timer Value Register

Address: 0x40000404, Reset: 0x0000, Name: T1VAL

Table 235. Bit Descriptions for T1VAL

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

Control Register

Address: 0x40000408, Reset: 0x000A, Name: T1CON

Table 236. Bit Descriptions for T1CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	EVENTEN	Event select. This bit enables and disables the capture of events. Used in conjunction with the EVENT select range: when a selected event occurs the current value of the up/down counter is captured in T1CAP. 0: events are not captured. 1: events are captured.	0x0	RW
[11:8]	EVENT	Event select range. Timer event select range (0 to 15).	0x0	RW
7	RLD	Reload control. RLD is only used for periodic mode; this bit allows the user to select whether the up/down counter is reset only on a timeout event or also when T1CLRI[0] is set. 1: up/down counter is reset the when T1CLRI[0] is set. 0: up/down counter is only reset on a timeout event.	0x0	RW
[6:5]	CLK	Clock select. These bits select a timer clock from the four available clock sources. 00: PCLK. 01: HCLK. 10: LFOSC, 32 kHz oscillator. 11: HFXTAL, if CLKCON0[11] = 1. 11: HFOSC, if CLKCON0[11] = 0.	0x0	RW

Bits	Bit Name	Description	Reset	Access
4	ENABLE	Timer enable. This bit enables and disables the timer. Clearing this bit resets the timer, including the T1VAL register. 0: DIS. Timer is disabled (default). 1: EN. Timer is enabled.	0x0	RW
3	MOD	Timer mode. This bit controls whether the timer runs in periodic or free running mode. In periodic mode, the up/down counter starts at the defined LOAD value (T1LD); in free running mode, the up/down counter starts at 0x0000 or 0xFFFF depending on whether the timer is counting up or down. 0: FREERUN. Timer runs in free running mode. 1: PERIODIC. Timer runs in periodic mode (default).	0x1	RW
2	UP	Count up. This bit control whether the timer increments (counts up) or decrements (counts down) the up/down counter. 0: DIS. Timer is set to count down (default). 1: EN. Timer is set to count up.	0x0	RW
[1:0]	PRE	Prescaler. These bits control the prescaler division factor applied to the selected clock of the timer. If CLK Source 0 or CLK Source 1 are selected, the prescaler value of 0 means divide by 4; otherwise, it means divide by 1. 00: Source Clock/[1 or 4]. 01: Source Clock/16. 10: Source Clock/256. 11: Source Clock/32,768.	0x2	RW

Clear Interrupt Register

Address: 0x4000040C, Reset: 0x0000, Name: T1CLRI

Table 237. Bit Descriptions for T1CLRI

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

Capture Register

Address: 0x40000410, Reset: 0x0000, Name: T1CAP

Table 238. Bit Descriptions for T1CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	CAP	16-bit captured value. T1CAP holds its value until T1CLRI[1] is set by user code. T1CAP is not overwritten even if another event occurs without writing to T1CLRI[1].	0x0	R

Status Register

Address: 0x4000041C, Reset: 0x0000, Name: T1STA

Table 239. Bit Descriptions for T1STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	PDOK	T1CLRI synchronization. This bit is set automatically when the user sets T1CLRI[0] = 1. It is cleared automatically when the clear interrupt request crosses clock domains and takes effect in the timer clock domain. 0: CLR. The interrupt is cleared in the timer clock domain. 1: SET. T1CLRI[0] is being updated in the timer clock domain.	0x0	R
6	BUSY	Timer busy. This bit informs the user that a write to T1CON is still crossing into the timer clock domain. Check this bit after writing T1CON, and suppress further writes until this bit is cleared. 0: CLR. Timer ready to receive commands to T1CON. 1: SET. Timer not ready to receive commands to T1CON.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: CLR. No capture event is pending. 1: SET. A capture event is pending.	0x0	R
0	TMOUT	Timeout event occurred. This bit is set automatically when the value of the counter reaches zero while counting down or reaches full scale when counting up. This bit is cleared when T1CLRI[0] is set by the user. 0: CLR. No timeout event has occurred. 1: SET. A timeout event has occurred.	0x0	R

REGISTER SUMMARY: GENERAL-PURPOSE TIMER 2

Table 240. Timer 2 Register Summary

Address	Name	Description	Reset	RW
0x40000800	T2LD	16-bit load value register	0x0000	RW
0x40000804	T2VAL	16-bit timer value register	0x0000	R
0x40000808	T2CON	Control register	0x000A	RW
0x4000080C	T2CLRI	Clear interrupt register	0x0000	W
0x40000810	T2CAP	Capture register	0x0000	R
0x4000081C	T2STA	Status register	0x0000	R

REGISTER DETAILS: GENERAL-PURPOSE TIMER 2

16-Bit Load Value Register

Address: 0x40000800, Reset: 0x0000, Name: T2LD

Table 241. Bit Descriptions for T2LD

Bits	Bit Name	Description	Reset	Access
[15:0]	LOAD	Load value. The up/down counter is periodically loaded with this value if periodic mode is selected (T2CON[3] = 1). LOAD writes during up/down counter timeout events are delayed until the event has passed.	0x0	RW

16-Bit Timer Value Register

Address: 0x40000804, Reset: 0x0000, Name: T2VAL

Table 242. Bit Descriptions for T2VAL

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

Control Register

Address: 0x40000808, Reset: 0x000A, Name: T2CON

Table 243. Bit Descriptions for T2CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	EVENTEN	Event select. This bit enables and disables the capture of events. Used in conjunction with the EVENT select range: when a selected event occurs the current value of the up/down counter is captured in T2CAP. 0: events are not captured. 1: events are captured.	0x0	RW
[11:8]	EVENT	Event select range. Timer event select range (0 to 15).	0x0	RW
7	RLD	Reload control. RLD is only used for periodic mode; this bit allows the user to select whether the up/down counter is reset only on a timeout event or also when T2CLRI[0] is set. 1: up/down counter is reset when T2CLRI[0] is set. 0: up/down counter is only reset on a timeout event.	0x0	RW
[6:5]	CLK	Clock select. These bits select a timer clock from the four available clock sources. 00: PCLK. 01: HCLK. 10: LFOSC, 32 kHz oscillator. 11: HFXTAL, if CLKCON0[11] = 1. 11: HFOSC, if CLKCON0[11] = 0.	0x0	RW

Bits	Bit Name	Description	Reset	Access
4	ENABLE	Timer enable. This bit enables and disables the timer. Clearing this bit resets the timer, including the T2VAL register. 0: DIS. Timer is disabled (default). 1: EN. Timer is enabled.	0x0	RW
3	MOD	Timer mode. This bit controls whether the timer runs in periodic or free running mode. In periodic mode, the up/down counter starts at the defined LOAD value (T2LD); in free running mode, the up/down counter starts at 0x0000 or 0xFFFF depending on whether the timer is counting up or down. 0: FREERUN. Timer runs in free running mode. 1: PERIODIC. Timer runs in periodic mode (default).	0x1	RW
2	UP	Count up. This bit controls whether the timer increments (counts up) or decrements (counts down) the up/down counter. 0: DIS. Timer is set to count down (default). 1: EN. Timer is set to count up.	0x0	RW
[1:0]	PRE	Prescaler. These bits control the prescaler division factor applied to the selected clock of the timer. If CLK Source 0 or CLK Source 1 are selected, the prescaler value of 0 means divide by 4; otherwise, it means divide by 1. 00: Source Clock/[1 or 4]. 01: Source Clock/16. 10: Source Clock/256. 11: Source Clock/32,768.	0x2	RW

Clear Interrupt Register

Address: 0x4000080C, Reset: 0x0000, Name: T2CLRI

Table 244. Bit Descriptions for T2CLRI

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

Capture Register

Address: 0x40000810, Reset: 0x0000, Name: T2CAP

Table 245. Bit Descriptions for T2CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	CAP	16-bit captured value. T2CAP holds its value until T2CLRI[1] is set by user code. T2CAP is not overwritten even if another event occurs without writing to T2CLRI[1].	0x0	R

Status Register

Address: 0x4000081C, Reset: 0x0000, Name: T2STA

Table 246. Bit Descriptions for T2STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	PDOK	T2CLRI synchronization. This bit is set automatically when the user sets T2CLRI[0] = 1. It is cleared automatically when the clear interrupt request crosses clock domains and takes effect in the timer clock domain. 0: CLR. The interrupt is cleared in the timer clock domain. 1: SET. T2CLRI[0] is being updated in the timer clock domain.	0x0	R
6	BUSY	Timer busy. This bit informs the user that a write to T2CON is still crossing into the timer clock domain. Check this bit after writing T2CON and suppress further writes until this bit is cleared. 0: CLR. Timer ready to receive commands to T2CON. 1: SET. Timer not ready to receive commands to T2CON.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: CLR. No capture event is pending. 1: SET. A capture event is pending.	0x0	R
0	TMOUT	Timeout event occurred. This bit set automatically when the value of the counter reaches zero while counting down or reaches full scale when counting up. This bit is cleared when T2CLRI[0] is set by the user. 0: CLR. No timeout event has occurred. 1: SET. A timeout event has occurred.	0x0	R

WATCHDOG TIMER

WATCHDOG TIMER FEATURES

The watchdog timer is a 16-bit count-down timer, which can recover from an invalid software state. The watchdog timer is clocked by the 32 kHz internal oscillator (LFOSC) with a programmable prescaler (1, 16,256, or 4096).

WATCHDOG TIMER BLOCK DIAGRAM

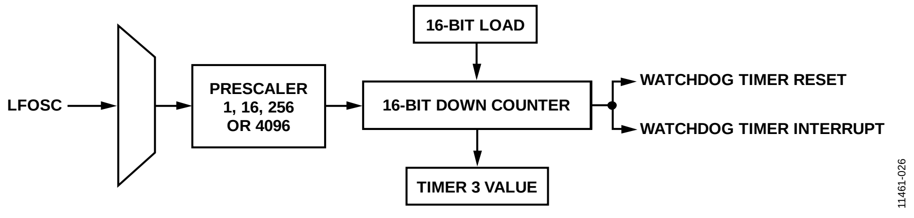


Figure 28. Watchdog Timer Block Diagram

WATCHDOG TIMER OVERVIEW

The watchdog timer (Timer 3) recovers from an invalid software state. When enabled, the watchdog timer requires periodic servicing to prevent it from forcing a device reset. For debug purposes, the timer can be configured to generate an interrupt instead of a reset.

The watchdog timer is clocked by the internal 32.768 kHz oscillator, LFOSC. It is clocked at all times except during a reset.

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

WATCHDOG TIMER OPERATION

The watchdog timer is enabled by default after a reset.

User code must disable the watchdog timer at the start of user code when debugging or if the watchdog timer is not required.

```
T3CON = 0x00;                                    // Disable watchdog timer
```

Enabling the watchdog timer (set T3CON[5] = 1) also write protects T3CON and T3LD.

This means that after kernel execution, user code can disable the timer and then reconfigure it with T3CON[5] = 1 only once. Then T3CON and T3LD are write protected. T3STA[4] indicates if the timer configuration has been locked. Only a reset clears T3CON[5], unlocking T3CON and T3LD, and allows reconfiguration of the timer.

If T3CON is not modified, user code can change T3LD at any time. If T3CON[5] is cleared to 0, the timer is disabled. Settings can be modified, and the timer can be reenabled.

When the watchdog timer is used in interrupt mode, T3STA[0], the watchdog timer interrupt bit, is only set to 1 for a very short period (2 × PCLK). Therefore, T3STA[0] must not be used for polling purposes.

REGISTER SUMMARY: WATCHDOG TIMER

Table 247. Watchdog Timer Register Summary

Address	Name	Description	Reset	Access
0x40002580	T3LD	Load value register	0x1000	RW
0x40002584	T3VAL	Current count value register	0x1000	R
0x40002588	T3CON	Control register	0x00E9	RW
0x4000258C	T3CLRI	Clear interrupt register	0x0000	W
0x40002598	T3STA	Status register	0x0000	R

REGISTER DETAILS: WATCHDOG TIMER

Load Value Register

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

Table 248. Bit Descriptions for T3LD

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

Current Count Value Register

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

Table 249. Bit Descriptions for T3VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	CCOUNT	Current count value	0x1000	R

Control Register

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

Table 250. Bit Descriptions for T3CON

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x1	R
6	MOD	Timer mode. Note that in free running mode, it wraps around at 0x1000. 0: FREERUN. Cleared by user to operate in free running mode. 1: PERIODIC. Set by user to operate in periodic mode (default).	0x1	RW
5	ENABLE	Timer enable. 0: DIS. Cleared by user to disable the timer. 1: EN. Set by user to enable the timer (default).	0x1	RW
4	RESERVED	Reserved.	0x0	R
[3:2]	PRE	Prescaler. 00: DIV1. Source Clock/1. 01: DIV16. Source Clock/16. 10: DIV256. Source Clock/256 (default). 11: DIV4096. Source Clock/4096	0x2	RW
1	IRQ	Timer interrupt. 0: DIS. Cleared by user to generate a reset on a time out (default). 1: EN. Set by user to generate an interrupt when the timer times out. This feature is provided for debug purposes and is only available in active mode.	0x0	RW
0	PMD	Power mode disable. PMD controls the behavior of the watchdog when in hibernate mode. If the application requires prolonged periods of time spent in hibernate mode and it is not desirable to periodically wake up to service the watchdog timer, the counter within the watchdog timer can be suspended when entering the hibernate power mode. Regardless of how the PMD bit is set, it is recommended that the watchdog timer be cleared before entering hibernate mode. 0: DIS. The watchdog timer continues its count down while in hibernate mode. 1: EN. When hibernate mode is entered, the watchdog counter suspends its countdown. As hibernate mode is exited, the countdown resumes from its current count value (the count is not reset).	0x1	RW

Clear Interrupt Register

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

Table 251. Bit Descriptions for T3CLRI

Bits	Bit Name	Description	Reset	Access
[15:0]	CLRWDG	Clear watchdog. User writes 0xCCCC to reset/reload/restart Timer 3 or clear IRQ. A write of any other value causes a watchdog reset. This register is write only; it reads back 0. Do not write to this register if using the timer in IRQ mode.	0x0	W

Status Register

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

Table 252. Bit Descriptions for T3STA

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Reserved.	0x0	R
4	LOCK	Lock status bit. Set automatically in hardware if T3CON[5] has been set by user code. Cleared by default and until user code sets T3CON[5].	0x0	R
3	CON	T3CON write synchronization in progress. 0: internal bus and Timer 3 clock domains T3CON configuration values match. 1: internal bus T3CON register values are being synchronized to Timer 3 clock domain.	0x0	R
2	LD	T3LD write synchronization in progress. 0: internal bus and Timer 3 clock domains T3LD values match. 1: internal bus T3LD value is being synchronized to Timer 3 clock domain.	0x0	R
1	CLRI	T3CLRI write synchronization in progress. 0: internal bus T3CLRI write synchronization not complete. 1: internal bus T3CLRI write is being synchronized to Timer 3 clock domain. Timer 3 is restarted (if 0xCCCC was written) when sync is complete.	0x0	R
0	IRQ	Watchdog timer interrupt. 0: Timer 3 interrupt not pending. 1: Timer 3 interrupt pending.	0x0	R

WAKE-UP TIMER

WAKE-UP TIMER FEATURES

The wake-up timer has the following features:

- 32-bit counter (count down or count up)
- Four clock sources with programmable prescaler (1, 16,256, or 32,768)
  - Peripheral clock (PCLK)
  - 32 kHz internal oscillator (LFOSC)
  - External clock applied on Pin P1.0 (ECLKIN)
- Four compare points, one automatic increment

WAKE-UP TIMER BLOCK DIAGRAM

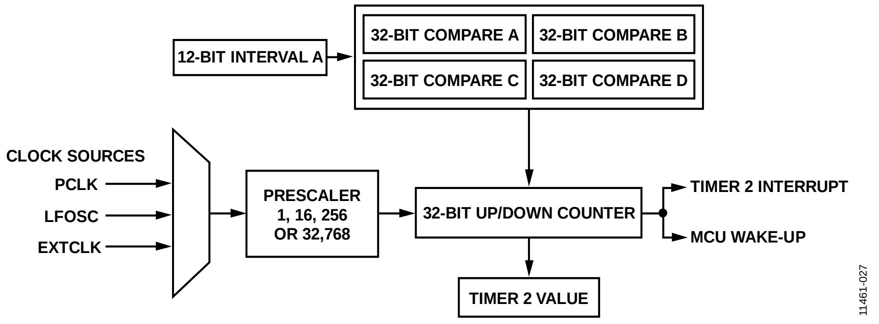


Figure 29. Wake-Up Timer Block Diagram

WAKE-UP TIMER OVERVIEW

The wake-up timer (Timer 4) block consists of a 32-bit counter clocked from one of three different sources: system clock (PCLK), internal oscillator (LFOSC), or an external clock applied on Pin P1.0 (ECLKIN). The selected clock source can be scaled down using a prescaler of 1, 16,256, or 32,768. The wake-up timer continues to run independent of the clock source used when the PCLK clock is disabled.

The timer can be used in free running or periodic mode. In free running mode, the timer counts from 0x00000000 to 0xFFFFFFFF and then restarts at 0x00000000. In periodic mode, the timer counts from 0x00000000 to T4WUFD (T4WUFD0 and T4WUFD1).

In addition, the wake-up timer has four specific time fields to compare with the wake-up counter: T4WUFA, T4WUFB, T4WUFC, and T4WUFD. All four wake-up compare points can generate interrupts or wake-up signals. When in free running mode, T4WUFA, T4WUFB, T4WUFC, andT4WUFD must be reconfigured in software to generate a periodic interrupt.

WAKE-UP TIMER OPERATION

The wake-up timer comparator registers must be configured before starting the timer. The timer is started by writing the control enable bit (T4CON[7]). The timer increments until the value reaches full scale in free running mode or when T4WUFD matches the wake-up value, T4VAL.

The wake-up timer is a 32-bit timer. Its current value is stored in two 16-bit registers: T4VAL1 stores the upper 16 bits, and T4VAL0 stores the lower 16 bits.

When T4VAL0 is read, T4VAL1 is frozen at the current value until it is subsequently read. The control bit, FREEZE (T4CON[3]), must be set to freeze the T4VAL register between the lower and upper reads.

Clock Selection

Clock selection is made by setting T4CON[10:9].

If PCLK is selected (T4CON[10:9] = 00), configuring T4CON[1:0] = 00 results in a prescaler of 4.

Synchronization to the LFOSC clock domain is done automatically by hardware, and precautions concerning asynchronous clocks as described in Timer 0, Timer 1, and Timer 3 do not apply.

Compare Field Registers

Hardware Updated Field

T4INC is a 12-bit interval register that updates the compare value in T4WUFAx by hardware. When a new value is written in T4INC, Bits[16:5] of the internal 32-bit compare register (T4WUFAx) are loaded with the new T4INC value. If the new compare value is less than the T4WUFD value in periodic mode or less than 0xFFFFFFFF in free running mode, this 32-bit compare register is automatically incremented with the contents of T4INC (shifted by five) each time the wake-up counter reaches the value in this compare register. If the new compare value is greater than these limits, it is recalculated as follows.

In free running mode, the new T4WUFA = old T4WUFA + (32 × T4INC) − 0xFFFFFFFF.

In periodic mode, the new T4WUFA = old T4WUFA + (32 × T4INC) − T4WUFD.

The maximum programmable interval is just above 4 seconds.

T4INC is compared with Bits[16:5] of the timer value. Because it is shifted left by 5 bits, the value must be multiplied by 32 to obtain the compare value.

With the default value of 0xC8 (where for calculation purposes 0xC8 = 200 in decimal), prescaler = 1, and 32 kHz clock selected,

$$Interval = ((200 \times 32) + 1) \times 1/32{,}768 = 195.3155 \text{ ms}$$

To modify the interval value, the timer must be stopped so that the interval register can be loaded in the compare register if T4CON[11] = 0.

To modify the interval value, set STOPINC (T4CON[11] = 1) while the timer is running.

The new T4INC value takes effect after the next Wake-Up Field A interrupt. If the user is writing to this register while the timer is enabled, set the STOPINC bit before writing to it, and then clear the STOPINC bit after the update.

Software Updated Field

T4WUFB, T4WUFC, and T4WUFD are 32-bit values programmed by the user in the T4WUFx0 and T4WUFx1 registers (x = B, C, or D). T4WUFD contains the load value when the wake-up timer is configured in periodic mode.

The T4WUFBx and T4WUFCx registers can be written to at any time, but the corresponding interrupt enable (T4IEN[1] or T4IEN[2]) must be disabled. After the register is updated, the interrupt can be reenabled.

In periodic mode, the T4WUFDx registers can be written to only when the timer is disabled. In free running mode, the T4WUFDx registers can be written to while the timer is running. Before doing so, the corresponding interrupt enable (T4IEN[3]) must be disabled. After the register is updated, the interrupt can be reenabled.

In free running mode, T4WUFB, T4WUFC, and T4WUFD can be written to at any time; however, the corresponding interrupt enable in the T4IEN register must be disabled. After the register is updated, the interrupt can be reenabled. In periodic mode, this is only applicable to T4WUFB and T4WUFC.

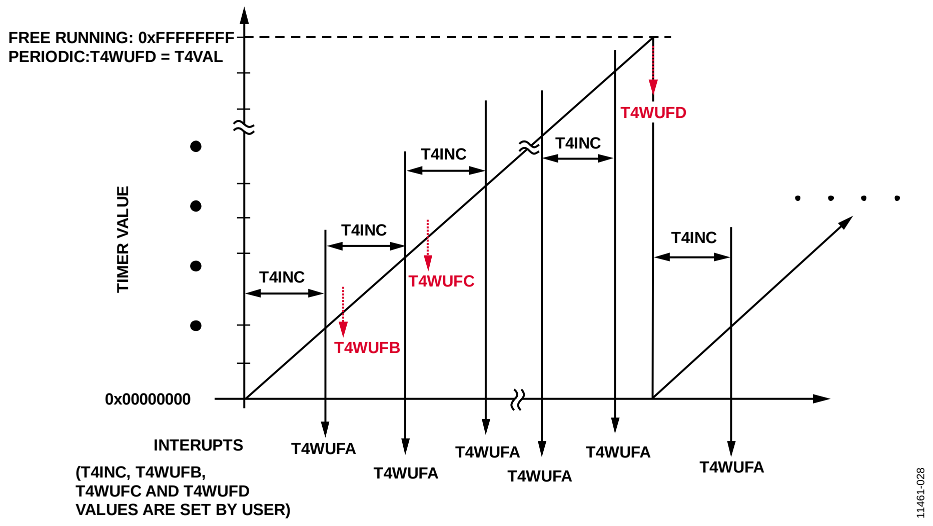


Figure 30. Wake-Up Timer Fields Action

Interrupts/Wake-Up Signals

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

The wake-up timer generates five maskable interrupts. They are enabled in the T4IEN register. Interrupts can be cleared by setting the corresponding bit in the T4CLRI register.

Note that it takes two 32 kHz clock cycles for the interrupt clear to take effect when the 32 kHz internal oscillator is used.

Ensure that the register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary. The following is a code example showing how to implement the DSB ARM Cortex-M3 instruction in a C program.

```
void Ext_Int4_Handler ()
{
   EiClr(EXTINT4);
   __DSB();
}
```

During that time, the device must not be placed in any of the power-down modes. IRQCRY (T4STA[6]) indicates when the device can be placed in power-down mode.

The timer is stopped and reset when clearing the timer enable bit in the T4CON register (T4CON[7]).

REGISTER SUMMARY: WAKE-UP TIMER

Table 253. Wake-Up Timer Register Summary

Address	Name	Description	Reset	Access
0x40002500	T4VAL0	Current count value—least significant 16 bits	0x0000	R
0x40002504	T4VAL1	Current count value—most significant 16 bits	0x0000	R
0x40002508	T4CON	Control register	0x0040	RW
0x4000250C	T4INC	12-bit interval for Wake-Up Field A	0x00C8	RW
0x40002510	T4WUFB0	Wake-Up Field B—least significant 16 bits	0x1FFF	RW
0x40002514	T4WUFB1	Wake-Up Field B—most significant 16 bits	0x0000	RW
0x40002518	T4WUFC0	Wake-Up Field C—least significant 16 bits	0x2FFF	RW
0x4000251C	T4WUFC1	Wake-Up Field C—most significant 16 bits	0x0000	RW
0x40002520	T4WUFD0	Wake-Up Field D—least significant 16 bits	0x3FFF	RW
0x40002524	T4WUFD1	Wake-Up Field D—most significant 16 bits	0x0000	RW
0x40002528	T4IEN	Interrupt enable register	0x0000	RW
0x4000252C	T4STA	Status register	0x0000	R
0x40002530	T4CLRI	Clear interrupt register	0x0000	W
0x4000253C	T4WUFA0	Wake-Up Field A—least significant 16 bits	0x1900	R
0x40002540	T4WUFA1	Wake-Up Field A—most significant 16 bits	0x0000	R

REGISTER DETAILS: WAKE-UP TIMER

Current Count Value—Least Significant 16 Bits Register

Address: 0x40002500, Reset: 0x0000, Name: T4VAL0

Table 254. Bit Descriptions for T4VAL0

Bits	Bit Name	Description	Reset	Access
[15:0]	T4VALL	Current count low. Least significant 16 bits of current count value.	0x0	R

Current Count Value—Most Significant 16 Bits Register

Address: 0x40002504, Reset: 0x0000, Name: T4VAL1

Table 255. Bit Descriptions for T4VAL1

Bits	Bit Name	Description	Reset	Access
[15:0]	T4VALH	Current count high. Most significant 16 bits of current count value.	0x0	R

Control Register

Address: 0x40002508, Reset: 0x0040, Name: T4CON

Table 256. Bit Descriptions for T4CON

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
11	STOP_WUFA	Disables updating Field A register T4WUFA. When set, this bit stops the Wake-Up Field A register T4WUFA from being updated with the interval register I2INC value. This allows the user to update the interval T4INC or T4WUFA registers safely.	0x0	RW
[10:9]	CLK	Clock select. 00: PCLK: peripheral clock (default). 01: LFOSC: 32 kHz internal oscillator. 10: LFOSC: 32kHz internal oscillator. 11: ECLKIN: external clock from P1.0.	0x0	RW
8	WUEN	Wake-up enable. 0: DIS: Cleared by user to disable the wake-up timer when the core clock is off. 1: EN: Set by user to enable the wake-up timer even when the core clock is off.	0x0	RW
7	ENABLE	Timer enable. 0: DIS: Disable the timer (default). 1: EN: Enable the timer.	0x0	RW
6	MOD	Timer mode. 0: PERIODIC: Cleared by user to operate in periodic mode. In this mode, the timer counts up to T4WUFD. 1: FREERUN: Set by user to operate in free running mode (default).	0x1	RW
[5:4]	RESERVED	Reserved. Write these bits 0.	0x0	RW
3	FREEZE	Freeze enable. 0: DIS: Cleared by user to disable this feature (default). 1: EN: Set by user to enable the freeze of the high 16 bits after the lower bits have been read from T4VAL0. This ensures that the software reads an atomic shot of the timer. T4VAL1 unfreezes after it has been read.	0x0	RW
2	RESERVED	Reserved.	0x0	RW
[1:0]	PRE	Prescaler. 00: PREDIV1: Source Clock/1 (default). If the selected clock source is PCLK, this setting results in a prescaler of 4. 01: PREDIV16: Source Clock/16. 10: PREDIV256: Source Clock/256. 11: PREDIV32768: Source Clock/32,768.	0x0	RW

12-Bit Interval for Wake-Up Field A Register

Address: 0x4000250C, Reset: 0x00C8, Name: T4INC

Table 257. Bit Descriptions for T4INC

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

Wake-Up Field B—Least Significant 16 Bits Register

Address: 0x40002510, Reset: 0x1FFF, Name: T4WUFB0

Table 258. Bit Descriptions for T4WUFB0

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFBL	Wake-Up Field B low. Least significant 16 bits of Wake-Up Field B.	0x1FFF	RW

Wake-Up Field B—Most Significant 16 Bits Register

Address: 0x40002514, Reset: 0x0000, Name: T4WUFB1

Table 259. Bit Descriptions for T4WUFB1

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFBH	Wake-Up Field B High. Most significant 16 bits of Wake-Up Field B.	0x0	RW

Wake-Up Field C—Least Significant 16 Bits Register

Address: 0x40002518, Reset: 0x2FFF, Name: T4WUFC0

Table 260. Bit Descriptions for T4WUFC0

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFCL	Wake-Up Field C Low. Least significant 16 bits of Wake-Up Field C.	0x2FFF	RW

Wake-Up Field C—Most Significant 16 Bits Register

Address: 0x4000251C, Reset: 0x0000, Name: T4WUFC1

Table 261. Bit Descriptions for T4WUFC1

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFCH	Wake-Up Field C High. Most significant 16 bits of Wake-Up Field C.	0x0	RW

Wake-Up Field D—Least Significant 16 Bits Register

Address: 0x40002520, Reset: 0x3FFF, Name: T4WUFD0

Table 262. Bit Descriptions for T4WUFD0

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFD0	Wake-Up Field D Low. Least significant 16 bits of Wake-Up Field C.	0x3FFF	RW

Wake-Up Field D—Most Significant 16 Bits Register

Address: 0x40002524, Reset: 0x0000, Name: T4WUFD1

Table 263. Bit Descriptions for T4WUFD1

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFDH	Wake-Up Field D high. Most significant 16 bits of Wake-Up Field D.	0x0	RW

Interrupt Enable Register

Address: 0x40002528, Reset: 0x0000, Name: T4IEN

Table 264. Bit Descriptions for T4IEN

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	R
4	ROLL	Rollover interrupt enable. Used only in free running mode. Set by user to generate an interrupt when Timer 2 rolls over. Cleared by user to disable the rollover interrupt (default).	0x0	RW
3	WUFD	T4WUFD interrupt enable. Set by user code to generate an interrupt when T4VAL reaches T4WUFD. Cleared by user code to disable T4WUFD interrupt (default).	0x0	RW
2	WUFC	T4WUFC interrupt enable. Set by user code to generate an interrupt when T4VAL reaches T4WUFC. Cleared by user code to disable T4WUFC interrupt (default).	0x0	RW
1	WUFB	T4WUFB interrupt enable. Set by user code to generate an interrupt when T4VAL reaches T4WUFB. Cleared by user code to disable T4WUFB interrupt (default).	0x0	RW
0	WUFA	T4WUFA interrupt enable. Set by user code to generate an interrupt when T4VAL reaches T4WUFA. Cleared by user code to disable T4WUFA interrupt (default).	0x0	RW

Status Register

Address: 0x4000252C, Reset: 0x0000, Name: T4STA

Table 265. Bit Descriptions for T4STA

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x0	R
8	PDOK	Enable bit synchronized. Indicates when a change in the enable bit is synchronized to the 32 kHz clock domain. It is set high when the enable bit (Bit 5) in the control register is set or cleared. It returns low when the change in the enable bit has been synchronized to the 32 kHz clock domain.	0x0	R
7	FREEZE	Timer value freeze. Set automatically to indicate that the value in T4VAL1 is frozen. Cleared by automatically when T4VAL1 is read.	0x0	R
6	IRQCRY	Wake-up status to power down. Set automatically when any of the interrupts are still set in the external crystal clock domain. Cleared automatically when the interrupts are cleared, allowing power-down mode. User code must wait for this bit to be cleared before entering power-down mode.	0x0	R
5	RESERVED	Reserved.	0x0	R
4	ROLL	Rollover interrupt flag. Used only in free running mode. Set automatically to indicate a rollover interrupt has occurred. Cleared automatically after a write to T4CLRI.	0x0	R
3	WUFD	T4WUFD interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T4CLRI.	0x0	R
2	WUFC	T4WUFC interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T4CLRI.	0x0	R
1	WUFB	T4WUFB interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T4CLRI.	0x0	R
0	WUFA	T4WUFA interrupt flag. Set automatically to indicate a comparator interrupt has occurred. Cleared automatically after a write to the corresponding bit in T4CLRI.	0x0	R

Clear Interrupt Register

Address: 0x40002530, Reset: 0x0000, Name: T4CLRI

Table 266. Bit Descriptions for T4CLRI

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x0	R
4	ROLL	Rollover interrupt clear. Used only in free running mode. Set by user code to clear a rollover interrupt flag. Cleared automatically after synchronization.	0x0	RW
3	WUFD	T4WUFD interrupt clear.	0x0	RW
2	WUFC	T4WUFC interrupt clear. Set by user code to clear a T4WUFC interrupt flag. Cleared automatically after synchronization.	0x0	RW
1	WUFB	T4WUFB interrupt clear. Set by user code to clear a T4WUFB interrupt flag. Cleared automatically after synchronization.	0x0	RW
0	WUFA	T4WUFA interrupt clear. Set by user code to clear a T4WUFA interrupt flag. Cleared automatically after synchronization.	0x0	RW

Wake-Up Field A—Least Significant 16 Bits Register

Address: 0x4000253C, Reset: 0x1900, Name: T4WUFA0

Table 267. Bit Descriptions for T4WUFA0

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFAL	Wake-Up Field A low. Least significant 16 bits of Wake-Up Field A.	0x1900	RW

Wake-Up Field A—Most Significant 16 Bits Register

Address: 0x40002540, Reset: 0x0000, Name: T4WUFA1

Table 268. Bit Descriptions for T4WUFA1

Bits	Bit Name	Description	Reset	Access
[15:0]	T4WUFAH	Wake-Up Field A high. Most significant 16 bits of Wake-Up Field A.	0x0	RW

PULSE WIDTH MODULATION (PWM)

PWM FEATURES

The ADuCM310 features an 8-channel PWM interface.

PWM OVERVIEW

The ADuCM310 integrates an 8-channel PWM interface. Eight channels are grouped as four pairs (0 to 3). The first two pairs of PWM outputs (PWM0 PWM1, PWM2, and PWM3) can be configured to drive an H-bridge. On power-up, the PWM outputs default to H-bridge mode. In standard mode, the user has control over the period of each pair of outputs and over the duty cycle of each individual output. The PWM trip interrupt can be cleared by writing 1 to PWMICLR[4]. When using the PWM trip interrupt, the PWM interrupt must be cleared before exiting the ISR, to prevent the generation of multiple interrupts.

Table 269. PWM Channel Grouping

Port Name	Description	PWM Mode Available
PWM0	High-side PWM output for Pair 0	H-bridge and standard
PWM1	Low-side PWM output for Pair 0	H-bridge and standard
PWM2	High-side PWM output for Pair 1	H-bridge and standard
PWM3	Low-side PWM output for Pair 1	H-bridge and standard
PWM4	High-side PWM output for Pair 2	Standard
PWM5	Low-side PWM output for Pair 2	Standard
PWM6	High-side PWM output for Pair 3	Standard
PWM7	Low-side PWM output for Pair 3	Standard

PWM OPERATION

In all modes, the PWMxCOMx MMRs control the point at which the PWM output changes state. The PWM clock is selectable via PWMCON0 with one of the following values: UCLK divided by 2, 4, 8, 16, 32, 64, 128, or 256.

The length of the PWM period is defined by PWMxLEN. Each pair has an associated counter.

The PWM waveforms are set by the count value of the 16-bit timer and the compare register contents.

The low-side waveform, PWM1, goes high when the timer count reaches PWM0LEN, and it goes low when the timer count reaches the value held in PWM0COM2 or when the high-side waveform PWM0 goes low.

The high-side waveform, PWM0, goes high when the timer count reaches the value held in PWM0COM0, and it goes low when the timer count reaches the value held in PWM0COM1.

Note that the high-side PWM output for each channel must have a high duration period greater than or equal to the high period duration of the low-side output. For example, the high period for PWM0 must be equal to or greater than the high period of PWM1.

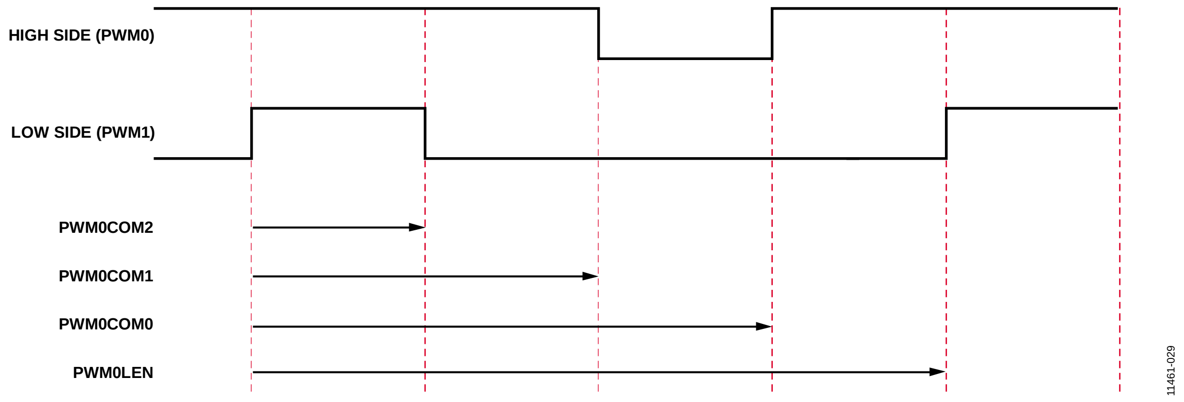


Figure 31. Waveform of PWM Channel Pair in Standard Mode

Table 270 lists equations for the period and duration for both the outputs of a PWM channel. Note that $t_{UCLK/DIV}$ is the PWM clock frequency selected by CLKCON1[2:0] and CLKSYSDIV[0], and $N_{PRESCALE}$ is the prescalar value as determined by PWMCON0[8:6].

Table 270. PWM Equations

PWM	Period	Low Duration
Low Side (PWM1)	$t_{UCLK/DIV}$ × (PWM0LEN + 1) × $N_{PRESCALE}$.	If (PWMCOM2 < PWMCOM1), $t_{UCLK/DIV}$ × (PWM0LEN − PWM0COM2) × $N_{PRESCALE}$. Else, t_{UCLK} × (PWM0LEN − PWM0COM1) × $N_{PRESCALE}$.
High Side (PWM0)	$t_{UCLK/DIV}$ × (PWM0LEN + 1) × $N_{PRESCALE}$.	$t_{UCLK/DIV}$ × (PWM0COM0 − PWM0COM1) × $N_{PRESCALE}$.

Standard Mode

In standard mode, each pair is individually controlled by a selection of registers, as shown in Table 271.

Table 271. Compare Register Descriptions in Standard Mode (Base Address: 0x40024000)

Name	Offset	Description
PWM0COM0	0x010	PWM0 output goes high when the PWM timer reaches the count value stored in this register.
PWM0COM1	0x014	PWM0 output goes low when the PWM timer reaches the count value stored in this register.
PWM0COM2	0x018	PWM1 output goes low when the PWM timer reaches the count value stored in this register.
PWM0LEN	0x01C	PWM1 output goes high when the PWM timer reaches the count value stored in this register.
PWM1COM0	0x020	PWM2 output goes high when the PWM timer reaches the count value stored in this register.
PWM1COM1	0x024	PWM2 output goes low when the PWM timer reaches the count value stored in this register.
PWM1COM2	0x028	PWM3 output goes low when the PWM timer reaches the count value stored in this register.
PWM1LEN	0x02C	PWM3 output goes high when the PWM timer reaches the count value stored in this register.
PWM2COM0	0x030	PWM4 output goes high when the PWM timer reaches the count value stored in this register.
PWM2COM1	0x034	PWM4 output goes low when the PWM timer reaches the count value stored in this register.
PWM2COM2	0x038	PWM5 output goes low when the PWM timer reaches the count value stored in this register.
PWM2LEN	0x03C	PWM5 output goes high when the PWM timer reaches the count value stored in this register.
PWM3COM0	0x040	PWM6 output goes high when the PWM timer reaches the count value stored in this register.
PWM3COM1	0x044	PWM6 output goes low when the PWM timer reaches the count value stored in this register.
PWM3COM2	0x048	PWM7 output goes low when the PWM timer reaches the count value stored in this register.
PWM3LEN	0x04C	PWM7 output goes high when the PWM timer reaches the count value stored in this register.

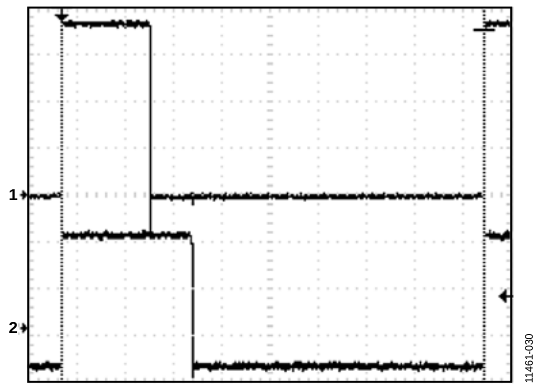


Figure 32. PWM Output on PWM0 and PWM1 Pins; PWM0 is Channel 2

H-Bridge Mode

In H-bridge mode, the two pairs of frequency and duty cycle are controlled by PWM0COM0, PWM0COM1, and PWM0LEN. For H-bridge mode, HMODE = 1 (PWMCON0[1] = 1). The HMODE bit also works with PWMCON0[5:2] for H-bridge mode. Note that only PWM0 to PWM3 participate in H-bridge mode; other outputs (PWM4 and PWM5) do not and continue to generate standard mode output.

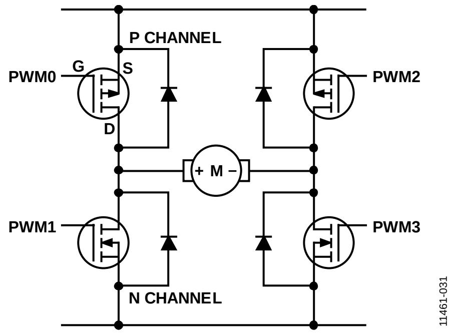


Figure 33. Example H-Bridge Configuration

Table 272. PWM Output in H-Bridge Mode

PWM Control Bits				PWM Outputs[1]				
ENA PWMCON0[9]	**POINV PWMCON0[5]**	**HOFF PWMCON0[4]**	**DIR PWMCON0[2]**	**PWM0**	**PWM1**	**PWM2**	**PWM3**	**State of Motor**
0	X	0	X	1 (Disable)	1 (Enable)	1	1	Brake
X	X	1	X	1 (Disable)	0 (Disable)	1	0	Free run
1	0	0	0	0 (Enable)	0 (Disable)	HS	LS	Move controlled by LS on PWM2
1	0	0	1	HS	LS	0	0	Move controlled by HS on PWM0
1	1	0	0	$\overline{\text{LS}}$	$\overline{\text{HS}}$	1	1	Move controlled by LS on PWM0
1	1	0	1	1 (Disable)	1 (Enable)	$\overline{\text{LS}}$	$\overline{\text{HS}}$	Move controlled by $\overline{\text{HS}}$ on PWM2

[1] HS is high side, LS is low side, $\overline{\text{HS}}$ is inverse of high side, and $\overline{\text{LS}}$ is inverse of low side, as programmed in the PWM0 registers.

PWM INTERRUPT GENERATION

PWM Trip Function Interrupt

When the PWM trip function is enabled (TRIPEN, PWMCON1[6]) and the PWM trip input signal goes low (falling edge), the PWM peripheral disables itself (PWMCON0[0] = 0). It also generates the PWM trip interrupt. The interrupt is cleared by setting PWMCLRI[4].

PWM Output Pairs Interrupts

In standard mode, each PWM pair has a dedicated interrupt: IRQPWM0, IRQPWM1, IRQPWM2, IRQPWM3. In H-bridge mode, only IRQPWM0 is available.

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 0 changes from PWM0LEN to 0, it also generates the IRQPWM0 interrupt. The interrupt is cleared by setting PWMCLRI[0].

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 1 changes from PWM1LEN to 0, it also generates the IRQPWM1 interrupt. The interrupt is cleared by setting PWMCLRI[1].

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 2 changes from PWM2LEN to 0, it also generates the IRQPWM2 interrupt. The interrupt is cleared by setting PWMCLRI[2].

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 3 changes from PWM3LEN to 0, it also generates the IRQPWM3 interrupt. The interrupt is cleared by setting PWMCLRI[3].

REGISTER SUMMARY: PWM

Table 273. PWM Register Summary

Address	Name	Description	Reset	Access
0x40024000	PWMCON0	PWM control register	0x0012	RW
0x40024004	PWMCON1	ADC conversion start and trip control register	0x0000	RW
0x40024008	PWMICLR	Hardware trip configuration register	0x0000	RW1C
0x40024010	PWM0COM0	Compare Register 0 for PWM0 and PWM1	0x0000	RW
0x40024014	PWM0COM1	Compare Register 1 for PWM0 and PWM1	0x0000	RW
0x40024018	PWM0COM2	Compare Register 2 for PWM0 and PWM1	0x0000	RW
0x4002401C	PWM0LEN	Period value register for PWM0 and PWM1	0x0000	RW
0x40024020	PWM1COM0	Compare Register 0 for PWM2 and PWM3	0x0000	RW
0x40024024	PWM1COM1	Compare Register 1 for PWM2 and PWM3	0x0000	RW
0x40024028	PWM1COM2	Compare Register 2 for PWM2 and PWM3	0x0000	RW
0x4002402C	PWM1LEN	Period value register for PWM2 and PWM3	0x0000	RW
0x40024030	PWM2COM0	Compare Register 0 for PWM4 and PWM5	0x0000	RW
0x40024034	PWM2COM1	Compare Register 1 for PWM4 and PWM5	0x0000	RW
0x40024038	PWM2COM2	Compare Register 2 for PWM4 and PWM5	0x0000	RW
0x4002403C	PWM2LEN	Period value register for PWM4 and PWM5	0x0000	RW
0x40024040	PWM3COM0	Compare Register 0 for PWM6 and PWM7	0x0000	RW
0x40024044	PWM3COM1	Compare Register 1 for PWM6 and PWM7	0x0000	RW
0x40024048	PWM3COM2	Compare Register 2 for PWM6 and PWM7	0x0000	RW
0x4002404C	PWM3LEN	Period value register for PWM6 and PWM7	0x0000	RW

REGISTER DETAILS: PWM

PWM Control Register

Address: 0x40024000, Reset: 0x0012, Name: PWMCON0

Table 274. Bit Descriptions for PWMCON0

Bits	Bit Name	Description	Reset	Access
15	SYNC	Set to enable PWM synchronization from the SYNC pin of the PWM. 0: ignore transition from the SYNC pin. 1: all PWM counters are reset on the next clock cycle after detection of a falling edge from the SYNC pin.	0x0	RW
14	PWM7INV	Set to invert PWM7 output.	0x0	RW
13	PWM5INV	Set to invert PWM5 output.	0x0	RW
12	PWM3INV	Set to invert PWM3 output.	0x0	RW
11	PWM1INV	Set to invert PWM1 output.	0x0	RW
10	PWMIEN	Set to enable interrupts for PWM.	0x0	RW
9	ENA	When HOFF = 0 and HMODE = 1, this bit serves as enable for Pair 0 and Pair 1. 0: disable Pair 0 and Pair 1. 1: enable Pair 0 and Pair 1.	0x0	RW
[8:6]	PWMCMP	PWM clock prescaler. Sets HCLK divider. 000: HCLK/2. 001: HCLK/4. 010: HCLK/8. 011: HCLK/16. 100: HCLK/32. 101: HCLK/64. 110: HCLK/128. 111: HCLK/256.	0x0	RW
5	POINV	Set to invert PWM outputs for Pair 0 and Pair 1 when PWM is in H-bridge mode.	0x0	RW
4	HOFF	Set to turn off the high-side for Pair 0 and Pair 1 when PWM is in H-bridge mode.	0x1	RW

Bits	Bit Name	Description	Reset	Access
3	LCOMP	Signal to load a new set of compare register values. In standard mode, this bit is cleared when the new values are loaded in the compare registers for all the channels. In H-bridge mode, this bit is not cleared; the user must write a value of 1 to this bit for the compare registers to be loaded. 0: use the values previously store in the compare and length registers. 1: load the internal compare registers with values stored in the PWMxCOMx and PWMxLEN registers.	0x0	RW
2	DIR	Direction control when PWM is in H-bridge mode. 0: PWM2 and PWM3 act as output signals while PWM0 and PWM1 are held low. 1: PWM0 and PWM1 act as output signals while PWM2 and PWM3 are held low.	0x0	RW
1	HMODE	Set to enable H-bridge mode.	0x1	RW
0	PWMEN	Master enable for PWM. 0: disable all PWM outputs. 1: enable all PWM outputs.	0x0	RW

ADC Conversion Start and Trip Control Register

Address: 0x40024004, Reset: 0x0000, Name: PWMCON1

Table 275. Bit Descriptions for PWMCON1

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved. Return 0 on reads.	0x00	Reserved
6	TRIP_EN	Set to enable PWM trip functionality.	0x0	RW
[5:0]	RESERVED	Reserved.	0x0	Reserved

Hardware Trip Configuration Register

Address: 0x40024008, Reset: 0x0000, Name: PWMICLR

Table 276. Bit Descriptions for PWMICLR

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved. Return 0 on reads.	0x000	Reserved
4	TRIP	Write a 1 to clear latched IRQPWMTrip interrupt. Returns 0 on reads.	0x0	RW1C
3	PWM3	Write a 1 to clear latched IRQPWM3 interrupt. Returns 0 on reads.	0x0	RW1C
2	PWM2	Write a 1 to clear latched IRQPWM2 interrupt. Returns 0 on reads.	0x0	RW1C
1	PWM1	Write a 1 to clear latched IRQPWM1 interrupt. Returns 0 on reads.	0x0	RW1C
0	PWM0	Write a 1 to clear latched IRQPWM0 interrupt. Returns 0 on reads.	0x0	RW1C

Compare Register 2 for PWM0 and PWM1

Address: 0x40024018, Reset: 0x0000, Name: PWM0COM2

Table 277. Bit Descriptions for PWM0COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	COM2	Compare Register 2 data	0x0	RW

Period Value Register for PWM0 and PWM1

Address: 0x4002401C, Reset: 0x0000, Name: PWM0LEN

Table 278. Bit Descriptions for PWM0LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	LEN	Period value	0x0	RW

Compare Register 0 for PWM2 and PWM3

Address: 0x40024020, Reset: 0x0000, Name: PWM1COM0

Table 279. Bit Descriptions for PWM1COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	COM0	Compare Register 0 data	0x0	RW

Compare Register 1 for PWM2 and PWM3

Address: 0x40024024, Reset: 0x0000, Name: PWM1COM1

Table 280. Bit Descriptions for PWM1COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	COM1	Compare Register 1 data	0x0	RW

Compare Register 2 for PWM2 and PWM3

Address: 0x40024028, Reset: 0x0000, Name: PWM1COM2

Table 281. Bit Descriptions for PWM1COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	COM2	Compare Register 2 data	0x0	RW

Period Value Register for PWM2 and PWM3

Address: 0x4002402C, Reset: 0x0000, Name: PWM1LEN

Table 282. Bit Descriptions for PWM1LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	LEN	Period value	0x0	RW

Compare Register 0 for PWM4 and PWM5

Address: 0x40024030, Reset: 0x0000, Name: PWM2COM0

Table 283. Bit Descriptions for PWM2COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	COM0	Compare Register 0 data	0x0	RW

Compare Register 1 for PWM4 and PWM5

Address: 0x40024034, Reset: 0x0000, Name: PWM2COM1

Table 284. Bit Descriptions for PWM2COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	COM1	Compare Register 1 data	0x0	RW

Compare Register 2 for PWM4 and PWM5

Address: 0x40024038, Reset: 0x0000, Name: PWM2COM2

Table 285. Bit Descriptions for PWM2COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	COM2	Compare Register 2 data	0x0	RW

Period Value Register for PWM4 and PWM5

Address: 0x4002403C, Reset: 0x0000, Name: PWM2LEN

Table 286. Bit Descriptions for PWM2LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	LEN	Period value	0x0	RW

Compare Register 0 for PWM6 and PWM7

Address: 0x40024040, Reset: 0x0000, Name: PWM3COM0

Table 287. Bit Descriptions for PWM3COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	COM0	Compare Register 0 data	0x0	RW

Compare Register 1 for PWM6 and PWM7

Address: 0x40024044, Reset: 0x0000, Name: PWM3COM1

Table 288. Bit Descriptions for PWM3COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	COM1	Compare Register 1 data	0x0	RW

Compare Register 2 for PWM6 and PWM7

Address: 0x40024048, Reset: 0x0000, Name: PWM3COM2

Table 289. Bit Descriptions for PWM3COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	COM2	Compare Register 2 data	0x0	RW

Period Value Register for PWM6 and PWM7

Address: 0x4002404C, Reset: 0x0000, Name: PWM3LEN

Table 290. Bit Descriptions for PWM3LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	LEN	Period value	0x0	RW

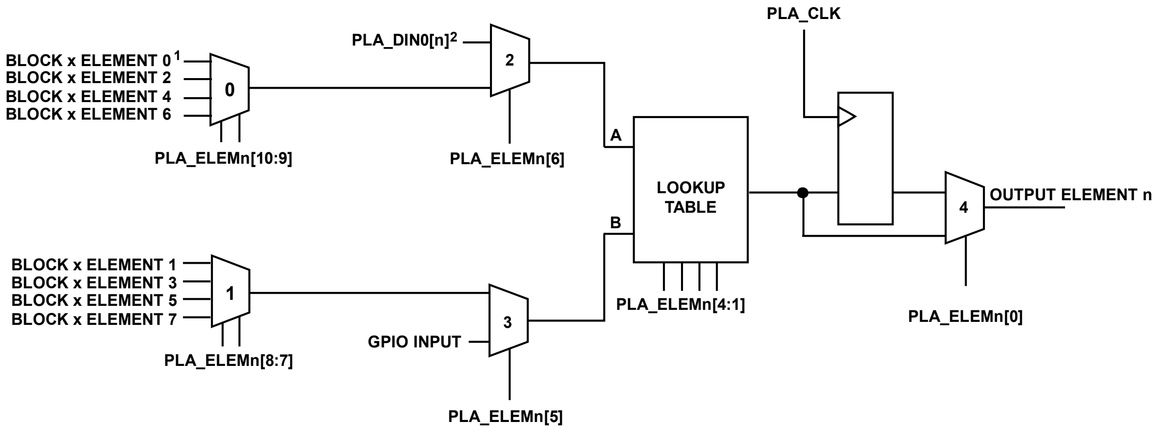
PROGRAMMABLE LOGIC ARRAY (PLA)

PLA FEATURES

The ADuCM310 integrates a fully programmable logic array (PLA) that consists of four independent but interconnected PLA blocks. Each block consists of eight PLA elements: Block x Element 0 to Block x Element 7, where x is the block number. Each ADuCM310 has 4 blocks, giving a total of 32 PLA elements: Element 0 to Element 31.

PLA OVERVIEW

Each PLA element contains a two-input lookup table that can be configured to generate any logic output function based on two inputs and a flip-flop.



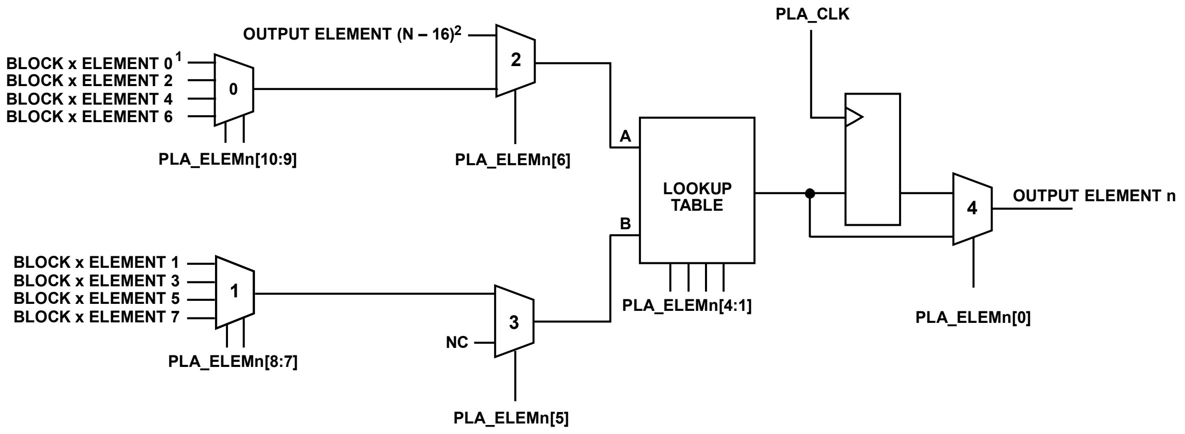
WHERE:
BLOCK x IS BLOCK 0 OR BLOCK 1
PLA_ELEMn IS THE MMR CONTROLLING ELEMENT n, n = 0 TO 15
NC = NO CONNECTION

[1]THE FIRST SELECTION OF MUX0 IS THE FEEDBACK FROM BLOCK x ELEMENT 0, WHERE x IS THE NUMBER OF THE CURRENT BLOCK. IF THE FIRST ELEMENT IN THE BLOCK IS BEING CONFIGURED, THE FEEDBACK COMES FROM ANOTHER BLOCK. SEE THE INTERBLOCK CONNECTION DIAGRAM FOR MORE DETAILS.

[2]FOR BLOCK 0 AND BLOCK 1 IS SET IN THE CORRESPONDING BIT IN THE PLA_DIN0 MMR.

11461-032

Figure 34. PLA Element: Block 0 and Block 1



WHERE:
BLOCK x IS BLOCK 2 OR BLOCK 3
PLA_ELEMn IS THE MMR CONTROLLING ELEMENT n, n = 16 TO 31
NC = NO CONNECTION

[1]THE FIRST SELECTION OF MUX0 IS THE FEEDBACK FROM BLOCK x ELEMENT 0, WHERE x IS THE NUMBER OF THE CURRENT BLOCK. IF THE FIRST ELEMENT IN THE BLOCK IS BEING CONFIGURED, THE FEEDBACK COMES FROM ANOTHER BLOCK. SEE THE INTERBLOCK CONNECTION DIAGRAM.

[2]FOR BLOCK 2 AND BLOCK 3, THE INPUT COMES FROM THE OUTPUT OF ELEMENT (n – 16), WHERE n IS THE NUMBER OF THE ELEMENT BEING CONFIGURED. FOR EXAMPLE, FOR ELEMENT 25 THE INPUT TO MUX 2 COMES FROM ELEMENT 9. THIS ALLOWS GPIO INPUTS TO BE INDIRECTLY CONNECTED TO ELEMENTS IN BLOCK 2 AND BLOCK 3.

11461-033

Figure 35. PLA Element: Block 2 and Block 3

In total, 27 GPIO pins are available on each ADuCM310 for the PLA. These include 14 input pins and 13 output pins, which must be configured in the GPxCON register as PLA pins before using the PLA.

PLA OPERATION

The PLA is configured via a set of user MMRs. The output(s) of the PLA can be routed to the internal interrupt system, to the PLA_DOUTx MMRs, or to any of the 13 PLA output pins.

The GPIO inputs to the PLA are always connected to their corresponding elements regardless of the setting in GPxCON. This means that a pin can be used as both a digital output from the ADuCM310 and an input to the PLA at the same time.

A PLA block can have several clock sources for its output flip-flops, or the flip-flops can be individually bypassed. All output flip-flops in the same block, if not bypassed, share the same clock source. The configuration of the clock sources can be found in the PLA clock select register.

Each PLA element in a block can be connected to other elements in the same block by configuring the output of MUX0 and MUX1. The configuration of these two multiplexers can be found in the PLA_ELEMn configuration register. A complete list of the possible connections are available in Table 292 and Table 293.

The four blocks can be interconnected as follows:

- Output of Element 7 (Block 0 Element 7) can be fed back to the Input 0 of Mux 0 of Element 8 (Block 1 Element 0).
- Output of Element 15 (Block 1 Element 7) can be fed back to Input 0 of Mux 0 of Element 16 (Block 2 Element 0).
- Output of Element 23 (Block 2 Element 7) can be fed back to the Input 0 of Mux 0 of Element 24 (Block 3 Element 0).
- Output of Element 31 (Block 3 Element 7) can be fed back to Input 0 of Mux 0 of Element 0 (Block 0 Element 0).

See Figure 36 for more information.

There are four interrupts available for the PLA. These interrupts can be configured to trigger on the output of any element using the PLA_IRQ0 and PLA_IRQ1 registers. The interrupts are active high; therefore, the interrupts continue triggering until the output of the element goes low or until the IRQ is disabled. If an active low interrupt is required, an extra element must be configured as an inverter and then the interrupt configured to monitor the output of this new element. If an edge triggered interrupt is required, two extra elements must be used and configured as an edge detector (($\overline{A}$) AND A).

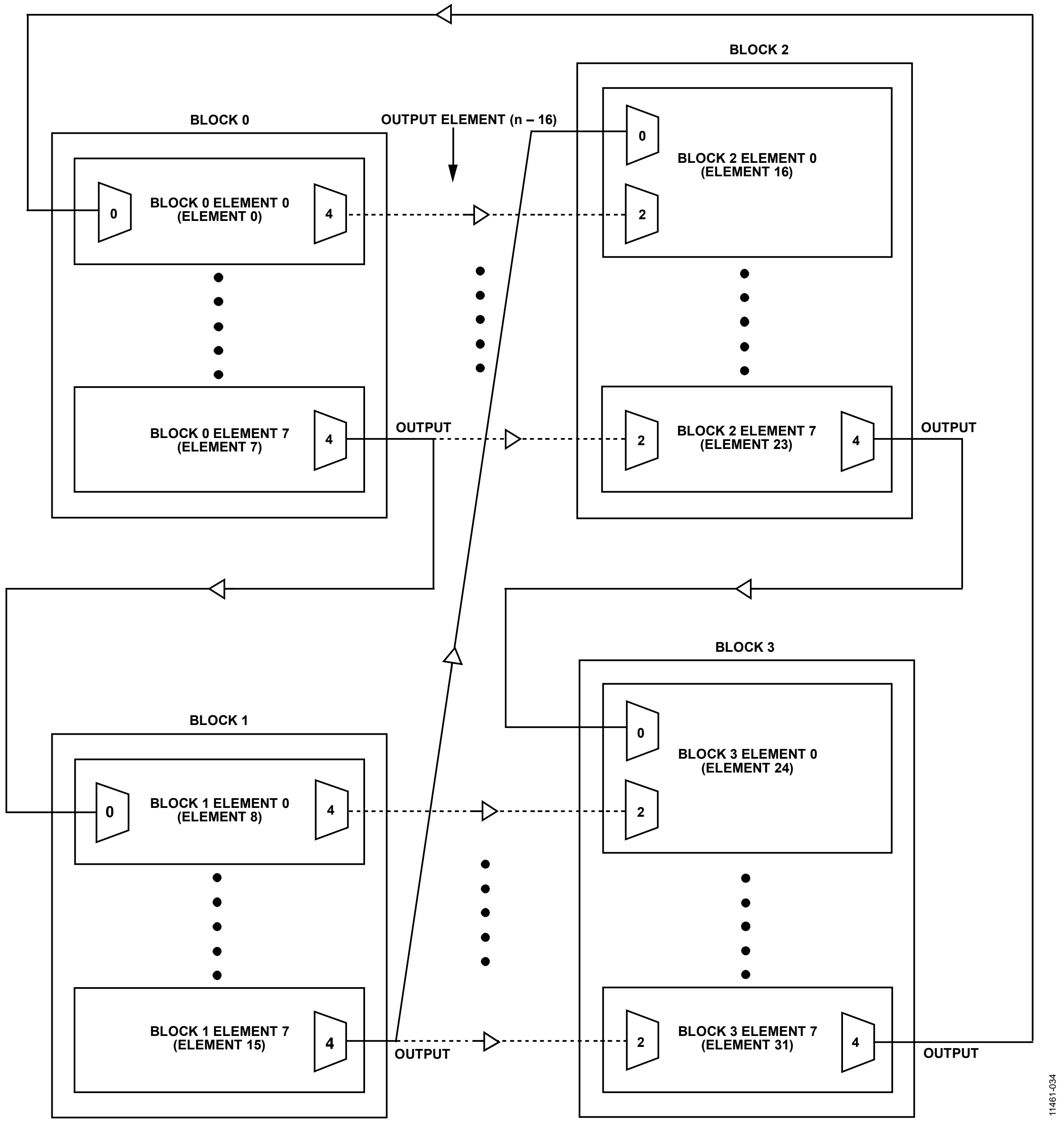


Figure 36. PLA Interblock Connections

Table 291. Element GPIO Input/Output

PLA Block 0			PLA Block 1			PLA Block 2		PLA Block 3	
Element	**Input**	**Output**	**Element**	**Input**	**Output**	**Element**	**Output**	**Element**	**Output**
0	P0.0		8	P2.0		16		24	
1	P0.1		9	P2.1		17		25	
2	P0.2	P0.4	10	P2.2	P1.4	18	P2.4	26	P3.4
3	P0.3	P0.5	11		P1.5	19	P2.5	27	
4	P1.0	P0.6	12	P3.0	P1.6	20	P2.6	28	
5	P1.1	P0.7	13	P3.1	P1.7	21	P2.7	29	
6	P1.2		14	P3.2		22		30	
7	P1.3		15			23		31	

Table 292. Mux 0 Feedback Configuration

PLA_ ELEMn[10:9]	PLA_ELEM0	PLA_ELEM1 to PLA_ELEM7	PLA_ELEM8	PLA_ELEM9 to PLA_ELEM15	PLA_ELEM16	PLA_ELEM17 to PLA_ELEM23	PLA_ELEM24	PLA_ELEM25 to PLA_ELEM31
00	Element 31	Element 0	Element 7	Element 8	Element 15	Element 16	Element 23	Element 24
01	Element 2	Element 2	Element 10	Element 10	Element 18	Element 18	Element 26	Element 26
10	Element 4	Element 4	Element 12	Element 12	Element 20	Element 20	Element 28	Element 28
11	Element 6	Element 6	Element 14	Element 14	Element 22	Element 22	Element 30	Element 30

Table 293. Mux 1 Feedback Configuration

PLA_ ELEMn[8:7]	PLA_ELEM0	PLA_ELEM1 to PLA_ELEM7	PLA_ELEM8	PLA_ELEM9 to PLA_ELEM15	PLA_ELEM16	PLA_ELEM17 to PLA_ELEM23	PLA_ELEM24	PLA_ELEM25 to PLA_ELEM31
00	Element 1	Element 1	Element 9	Element 9	Element 17	Element 17	Element 25	Element 25
01	Element 3	Element 3	Element 11	Element 11	Element 19	Element 19	Element 27	Element 27
10	Element 5	Element 5	Element 13	Element 13	Element 21	Element 21	Element 29	Element 29
11	Element 7	Element 7	Element 15	Element 15	Element 23	Element 23	Element 31	Element 31

Table 294. Lookup Table Configuration

PLA_ELEMn[4:1]	Function
0000	0
0001	A NOR B
0010	$\overline{A}$ AND B
0011	$\overline{A}$
0100	A AND $\overline{B}$
0101	$\overline{B}$
0110	A XOR B
0111	A NAND B
1000	A AND B
1001	A EXNOR B
1010	B
1011	$\overline{A}$ OR B
1100	A
1101	A OR $\overline{B}$
1110	A OR B
1111	1

REGISTER SUMMARY: PLA

Table 295. PLA Register Summary

Address	Name	Description	Reset	Access
0x40005800	PLA_ELEMn	ELEMx configuration register	0x0000	RW
0x40005880	PLA_CLK	PLA clock select	0x0000	RW
0x40005884	PLA_IRQ0	Interrupt register for Block 0 and Block 1	0x0000	RW
0x40005888	PLA_IRQ1	Interrupt register for Block 2 and Block 3	0x0000	RW
0x4000588C	PLA_ADC	ADC configuration register	0x0000	RW
0x40005890	PLA_DIN0	Data input for Block 0 and Block 1	0x0000	RW
0x40005898	PLA_DOUT0	Data output for Block 0 and Block 1	0x0000	R
0x4000589C	PLA_DOUT1	Data output for Block 2 and Block 3	0x0000	R
0x400058A0	PLA_LCK	Write lock register (can only be set once every reset)	0x0000	RW1S

REGISTER DETAILS: PLA

ELEMx Configuration Register

Address: 0x40005800 to 0x4000587C (Increments of 0x4), Reset: 0x0000, Name: PLA_ELEMn

Table 296. Bit Descriptions for PLA_ELEMn

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Not used.	0x00	Reserved
[10:9]	MUX0	Even element feedback selection (in respective block). 00: feedback from Element 0 (all except Element 0)/input from other blocks (Element 0 only). 01: feedback from Element 2. 10: feedback from Element 4. 11: feedback from Element 6.	0x0	RW
[8:7]	MUX1	Odd element feedback selection (in respective block). 00: feedback from Element 1. 01: feedback from Element 3. 10: feedback from Element 5. 11: feedback from Element 7.	0x0	RW
6	MUX2	Select between corresponding bit from PLA_DINx register or even feedback mux. 0: PLA_DINx input. 1: even feedback mux.	0x0	RW
5	MUX3	Select between GPIO bus input and odd feedback input (for Element 16 to Element 31, odd feedback is always selected). 0: odd feedback mux. 1: GPIO input.	0x0	RW
[4:1]	TBL	Bit 4, Bit 3, Bit 2, Bit 1 configures output for {mux2_out, mux3_out} = 11, 10, 01, 00, respectively. 0000: 0. 0001: NOR. 0010: B and not A. 0011: NOT A. 0100: A and not B. 0101: Not B. 0110: EXOR. 0111: NAND. 1000: AND. 1001: EXNOR. 1010: B. 1011: B or not A. 1100: A. 1101: A or not B. 1110: OR. 1111: 1.	0x0	RW

Bits	Bit Name	Description	Reset	Access
0	MUX4	Select or bypass flip-flop output. 0: flip-flop output. 1: bypass output	0x0	RW

PLA Clock Select Register

Address: 0x40005880, Reset: 0x0000, Name: PLA_CLK

Table 297. Bit Descriptions for PLA_CLK

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Not used.	0x0	Reserved
[14:12]	BLOCK3	Clock select for Block 3. 000: GPIO clock on P0.3. 001: GPIO clock on P1.1. 010: GPIO clock on P2.0. 011: HCLK. 100: MOSC. 101: Timer 0. 110: Timer 2. 111: KOSC.	0x0	RW
11	RESERVED	Not used.	0x0	Reserved
[10:8]	BLOCK2	Clock select for Block 2. 000: GPIO clock on P0.3. 001: GPIO clock on P1.1. 010: GPIO clock on P2.0. 011: HCLK. 100: MOSC. 101: Timer 0. 110: Timer 2. 111: KOSC.	0x0	RW
7	RESERVED	Not used.	0x0	Reserved
[6:4]	BLOCK1	Clock select for Block 1. 000: GPIO clock on P0.3. 001: GPIO clock on P1.1. 010: GPIO clock on P2.0. 011: HCLK. 100: MOSC. 101: Timer 0. 110: Timer 2. 111: KOSC.	0x0	RW
3	RESERVED	Not used.	0x0	Reserved
[2:0]	BLOCK0	Clock select for Block 0. 000: GPIO clock on P0.3. 001: GPIO clock on P1.1. 010: GPIO clock on P2.0. 011: HCLK. 100: MOSC. 101: Timer 0. 110: Timer 2. 111: KOSC.	0x0	RW

Interrupt Register for Block 0 and Block 1

Address: 0x40005884, Reset: 0x0000, Name: PLA_IRQ0

Table 298. Bit Descriptions for PLA_IRQ0

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Not used.	0x0	Reserved
12	IRQ1_EN	IRQ1 enable. 0: disable IRQ1 interrupt. 1: enable IRQ1 interrupt.	0x0	RW
[11:8]	IRQ1_SRC	IRQ1 source select (Element 0 to Element 15). The 4-bit value corresponds to the element number (for example, 1011 selects Element 11).	0x0	RW
[7:5]	RESERVED	Not used.	0x0	Reserved
4	IRQ0_EN	IRQ0 enable. 0: disable IRQ0 interrupt. 1: enable IRQ0 interrupt.	0x0	RW
[3:0]	IRQ0_SRC	IRQ0 source select (Element 0 to Element 15). The 4-bit value corresponds to the element number (for example, 1011 selects Element 11).	0x0	RW

Interrupt Register for Block 2 and Block 3

Address: 0x40005888, Reset: 0x0000, Name: PLA_IRQ1

Table 299. Bit Descriptions for PLA_IRQ1

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Not used.	0x0	Reserved
12	IRQ3_EN	IRQ3 enable. 0: disable IRQ3 interrupt. 1: enable IRQ3 interrupt.	0x0	RW
[11:8]	IRQ3_SRC	IRQ3 source select (Element 16 to Element 31). The element number corresponds to the 4-bit value + 16 (for example, 1011 selects Element 27).	0x0	RW
[7:5]	RESERVED	Not used.	0x0	Reserved
4	IRQ2_EN	IRQ2 enable. 0: disable IRQ2 interrupt. 1: enable IRQ2 interrupt.	0x0	RW
[3:0]	IRQ2_SRC	IRQ2 source select (Element 16 to Element 31). The element number corresponds to the 4-bit value + 16 (for example, 1011 selects Element 27).	0x0	RW

ADC Configuration Register

Address: 0x4000588C, Reset: 0x0000, Name: PLA_ADC

Table 300. Bit Descriptions for PLA_ADC

Bits	Bit Name	Description	Reset	Access
[15:6]	RESERVED	Not used.	0x000	Reserved
5	CONVST_EN	Bit to enable ADC start convert from PLA. 0: disable. 1: enable.	0x0	RW
[4:0]	CONVST_SRC	Element for ADC start convert source. The binary value corresponds to the element number. For example, Element 23 is 10111.	0x00	RW

Data Input for Block 0 and Block 1 Register

Address: 0x40005890, Reset: 0x0000, Name: PLA_DIN0

Table 301. Bit Descriptions for PLA_DIN0

Bits	Bit Name	Description	Reset	Access
[15:0]	DIN	Input bit to Element 15 to Element 0	0x0	RW

Data Output for Block 0 and Block 1 Register

Address: 0x40005898, Reset: 0x0000, Name: PLA_DOUT0

Table 302. Bit Descriptions for PLA_DOUT0

Bits	Bit Name	Description	Reset	Access
[15:0]	DOUT	Output bit from Element 15 to Element 0	0x0	R

Data Output for Block 2 and Block 3 Register

Address: 0x4000589C, Reset: 0x0000, Name: PLA_DOUT1

Table 303. Bit Descriptions for PLA_DOUT1

Bits	Bit Name	Description	Reset	Access
[15:0]	DOUT	Output bit from Element 31 to Element 16	0x0	R

Write Lock Register

Address: 0x400058A0, Reset: 0x0000, Name: PLA_LCK

This register can only be set once every reset.

Table 304. Bit Descriptions for PLA_LCK

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Not used.	0x0000	Reserved
0	LOCK	Set to disable writing to registers. 0: writing to registers allowed 1: writing to registers disabled	0x0	RW1S