CEC1712

Cryptographic Embedded Controller

Operating Conditions

- Operating Voltages: 3.3 V and 1.8 V
- Operating Temperature Range: -40 °C to 85 °C

Low Power Modes

- Chip is designed to always operate in Lowest Power state during Normal Operation
- Supports all 5 ACPI Power States for PC platforms
- Supports 2 Chip-level Sleep Modes: Light Sleep and Heavy Sleep
  - Low Standby Current in Sleep Modes

ARM® Cortex-M4 Embedded Processor

- Programmable clock frequency up to 48 MHz
- Fixed point processor
- Single 4GByte Addressing Space
- Nested Vectored Interrupt Controller (NVIC)
  - Maskable Interrupt Controller
  - Maskable hardware wake up events
  - 8 Levels of priority, individually assignable by vector
- EC Interrupt Aggregator expands number of Interrupt sources supported or reduces number of vectors needed
- Complete ARM® Standard debug support
  - JTAG-Based DAP port, comprised of SWJ-DP and AHB-AP debugger access functions

Memory Components

- 256 KB Code/Data SRAM
  - 224 KB optimized for code performance
  - 32 KB optimized for data performance
- 64 Bytes Battery Powered Storage SRAM
- 288 Bytes OTP
  - In circuit programmable
- ROM
  - Contains Boot ROM
  - Contains Runtime APIs for built-in functions

Clocks

- 48 MHz Internal PLL
- 32 kHz Clock Sources
  - Internal 32 kHz silicon oscillator
  - External 32 kHz crystal (XTAL) source
  - External single-ended 32 kHz clock source

Package Options

- 84 pin WFBGA

Security Features

- Boot ROM Secure Boot Loader
  - Hardware Root of trust using Secure Boot and Immutable code using ECDSA P-384 and SHA-384
  - Supports 2 Code Images in external SPI Flash (Primary and Fall back image)
  - Authenticates SPI Flash image before loading
  - Support AES-256 Encrypted SPI Flash images
  - Key Revocation
  - Roll back protection
  - DICE support
- Hardware Accelerators:
  - Multi purpose AES Crypto Engine:
    - Support for 128-bit - 256-bit key length
    - Supports Battery Authentication applications
  - Digital Signature Algorithm Support
    - Support for ECDSA and EC_KCDSA
  - Cryptographic Hash Engine
    - Support for SHA-1, SHA-256 to SHA-512
  - Public Key Crypto Engine
    - Hardware support for RSA and Elliptic Curve asymmetric public key algorithms
    - RSA keys length of 1024 to 4096 bits
    - ECC Prime Field keys up to 571 bits
    - ECC Binary Field keys up to 571 bits
    - Microcoded support for standard public key algorithms
  - OTP for storing Keys and IDs
    - Lockable on 32 B boundaries to prevent read access or write access
  - True Random Number Generator
  - 1 kbit FIFO
  - JTAG Disabled by default

Peripheral Features

- One Serial Peripheral Interface (SPI) Master Controller
  - Dual and Quad I/O Support
  - Flexible Clock Rates
  - Support for 1.8V and 3.3V slave devices
  - SPI Burst Capable
  - SPI Controller Operates with Internal DMA Controller with CRC Generation
  - Mappable to 2 ports (only 1 port active at a time)
  - SPI interface can be disabled after loading code
- Internal DMA Controller
  - Hardware or Firmware Flow Control
  - Firmware Initiated Memory-to-Memory transfers
  - Hardware CRC-32 Generator on Channel 0
  - 12-Hardware DMA Channels support five SMBus Master/Slave Controllers and One SPI Controller
- I2C/SMBus Controllers
  - 5 I2C/SMBus controllers
  - 3 I2C only controllers without the Network layer
  - 10 Configurable I2C ports
    - Full Crossbar switch allows any port to be connected to any controller
  - Supports Promiscuous mode of operation
  - Fully Operational on Standby Power
  - Multi-Master Capable
  - Supports Clock Stretching
  - Programmable Bus Speeds
  - 1 MHz Capable
  - Supports DMA Network Layer
- General Purpose I/O Pins
  - Inputs
    - Asynchronous rising and falling edge wakeup detection Interrupt High or Low Level
  - Outputs:
    - Push Pull or Open Drain output
    - Programmable power well emulation
  - Pull up or pull down resistor control
    - Automatically disabling pull-up resistors when output driven low
    - Automatically disabling pull-down resistors when output driven high
  - Programmable drive strength
  - Two separate1.8V/3.3V configurable IO regions
  - Group or individual control of GPIO data
  - 8 - Over voltage tolerant GPIO pins
  - Glitch protection and Under-Voltage Protection on all GPIO pins
- Input Capture and Compare timer
  - Six 32-bit Capture Registers
  - 11 Input Pins (ICTx)
    - Full Crossbar switch allows any port to be connected to any capture register
  - 32-bit Free-running timer
  - One 32-bit Compare Register output
  - Capture, Compare and Overflow Interrupts
- Universal Asynchronous Receiver Transmitter (UART)
  - Three High Speed NS16C550A Compatible UARTs with Send/Receive 16-Byte FIFOs
    - UART1 - Configurable 2-pin/4-pin
    - UART2 - 2-pin
    - UART3 - 2-pin
  - Programmable Main Power or Standby Power Functionality
  - Standard Baud Rates to 115.2 Kbps, Custom Baud Rates to 1.5 Mbps
- Programmable Timer Interface
  - Two16-bit Auto-reloading Timer Instances
    - 16 bit Pre-Scale divider
    - Halt and Reload control
    - Auto Reload
  - Two 32-bit Auto-reloading Timer Instances
    - 16 bit Pre-Scale divider
    - Halt and Reload control
    - Auto Reload
  - Three Operating Modes per Instance: Timer (Reload or Free-Running) or One-shot.
    - Event Mode is not supported
- 32-bit RTOS Timer
  - Runs Off 32kHz Clock Source
  - Continues Counting in all the Chip Sleep States regardless of Processor Sleep State
  - Counter is Halted when Embedded Controller is Halted (e.g., JTAG debugger active, break points)
  - Generates wake-capable interrupt event
- Watch Dog Timer (WDT)
  - Generates an interrupt prior to resetting
- 6 Programmable Pulse Width Modulator (PWM) outputs
    - Multiple Clock Rates
    - 16-Bit ON & 16-Bit OFF Counters
- 2 Fan Tachometer Inputs
    - 16 Bit Resolution
- Breathing LED Interface
  - Two Blinking/Breathing LEDs

  - Programmable Blink Rates
  - Piecewise Linear Breathing LED Output Controller
    - Provides for programmable rise and fall waveforms
  - Operational in EC Sleep States
  - Both 5V tolerant LED pins

Analog Features

- ADC Interface
  - 10-bit or 12-bit readings supported
  - ADC Conversion time 500nS/channel
  - 5 Channels
  - External voltage reference
  - Supports thermistor temperature readings

Battery Powered Peripherals

- Real Time Clock (RTC)
  - VBAT Powered
  - 32KHz Crystal Oscillator orExternal single-ended 32 kHz clock source
  - Time-of-Day and Calendar Registers
  - Programmable Alarms
  - Supports Leap Year and Daylight Savings Time
- Hibernation Timer Interface
  - Two 32.768 KHz Driven Timers
  - Programmable Wake-up from 0.5ms to 128 Minutes
- Week Timer
  - System Power Present Input Pin
    - Week Alarm Event only generated when System Power is Available
  - Power-up Event
  - Week Alarm Interrupt with 1 Second to 8.5 Year Time-out
  - Sub-Week Alarm Interrupt with 0.50 Seconds - 72.67 hours time-out
  - 1 Second and Sub-second Interrupts
- VBAT-Powered Control Interface (VCI)
  - 2 Active-low VCI Inputs
  - System Power Present Detection for gating RTC wake events
  - Optional filter
- Battery- powered General purpose Output (BGPO)

Debug Features

- 2-pin Serial Wire Debug (SWD) interface
- 4-Pin JTAG interface for Boundary Scan
- Trace FIFO Debug Port (TFDP)

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip’s Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

Table Of Contents

1.0 GENERAL DESCRIPTION

The CEC1712 device is a low power integrated embedded controller designed with strong cryptographic support. The CEC1712 is a highly-configurable, mixed-signal, advanced I/O controller architecture. It contains a 32-bit ARM® Cortex-M4 processor core with closely-coupled memory for optimal code execution and data access. An internal ROM, embedded in the design, is used to store the power on/boot sequence and APIs available during run time. When VTR_CORE is applied to the device, the secure boot loader API is used to download the custom firmware image from the system's shared SPI Flash device, thereby allowing system designers to customize the device's behavior.

The CEC1712 device is directly powered by a minimum of two separate suspend supply planes (VBAT and VTR). The CEC1712 has two banks of I/O pins that are able to operate at either 3.3 V or 1.8 V. Operating at 1.8V allows the CEC1712 to interface with the latest platform controller hubs and will lower the overall power consumed by the device, Whereas 3.3V allows this device to be integrated into legacy platforms that require 3.3V operation.

The CEC1712 secure boot loader authenticates and optionally decrypts the SPI Flash OEM boot image using the AES-256, ECDSA P-384, SHA-384 cryptographic hardware accelerators. The CEC1712 hardware accelerators support 128-bit and 256-bit AES encryption, ECDSA and EC_KCDSA signing algorithms, 1024-bits to 4096-bits RSA and Elliptic asymmetric public key algorithms, and a True Random Number Generator (TRNG). Runtime APIs are provided in the ROM for customer application code to use the cryptographic hardware. Additionally, the device offers lockable OTP storage for private keys and IDs. Additional features supported include Key Revocation, Roll back protection and DICE.

CEC1712 offers a software development system interface that includes a Trace FIFO debug port and a 2-pin Serail wire debug (SWD)/ JTAG interface.

1.1 Family Features

TABLE 1-1: CEC1712 FEATURE LIST

Features	CEC1712 -84 WFBGA
Package	84 pin WFBGA
Device ID	0023_A2
Boundary Scan JTAG ID	0223_2445
CPU	32-bit ARM® Cortex-M4
SRAM	256 kB
Code/Data Options (Primary use)	224kB/32kB
Battery Backed SRAM	64 bytes
Trace FIFO Debug Port	Yes
Internal DMA Channels	12
32-bit Timer	2
16-bit Timer	2
Capture Timer Registers	6
ICT Channels	11
Compare Timer	Yes
Watchdog Timer (WDT)	1
Hibernation Timer	2
Week Timer	1
Sub Week Timer	1
RTC	1
RTOS Timer	1
Keyboard Matrix scan support	No
SMBus 2.0 Host Controllers	5
I2C Host Controllers	3

TABLE 1-1: CEC1712 FEATURE LIST (CONTINUED)

Features	CEC1712 -84 WFBGA
Package	84 pin WFBGA
I2C/SMBus Ports	10
QMSPI Controller	1 Controller/2 ports
PWMs	6
Tachometers (TACHs)	2
GPIOs	68
Over voltage protected Pads	8
10/12- bit ADC Channels	5
UARTs	3 UART0: 2/4 pin configurable UART1: 2 pin UART2: 2 pin
Battery powered GPIO	1
VBAT powered Control Interface inputs	2
2 Pin parallel XTAL Oscillator	Yes
Single ended external 32kHz clock input (XTAL2)	Yes
JTAG	4-pin/2-pin
AES Hardware Support	128-256 bit
SHA Hashing Support	SHA-1 to SHA-512
Public Key Cryptography Support	RSA: 4K bit ECC: 571 bit
True Random Number Generator	1K bit
Root of Trust	Yes
Secure Boot	Yes
Immutable Code	Yes
Customer OTP	288 bytes
Optional OTP Selectable Features (Note 1)	
QA Testing	Yes
JTAG Disable	Yes
Authentication	Yes
Encrypt ECDH Private Key (Bytes 0-31)	Yes
AES Encryption Mandatory	Yes
OTP Write Lock - [0] ECDH Private Key	Yes
OTP Write Lock - [4] Authentication Key - Public Qx	Yes
OTP Write Lock - [5] Authentication Key - Public Qy	Yes
OTP Write Lock - [6] ECDH Public Key 2, Public Rx	Yes
OTP Write Lock - [7] ECDH Public Key 2, Public Ry	Yes
TAG0 SPI Flash Base Address	Yes

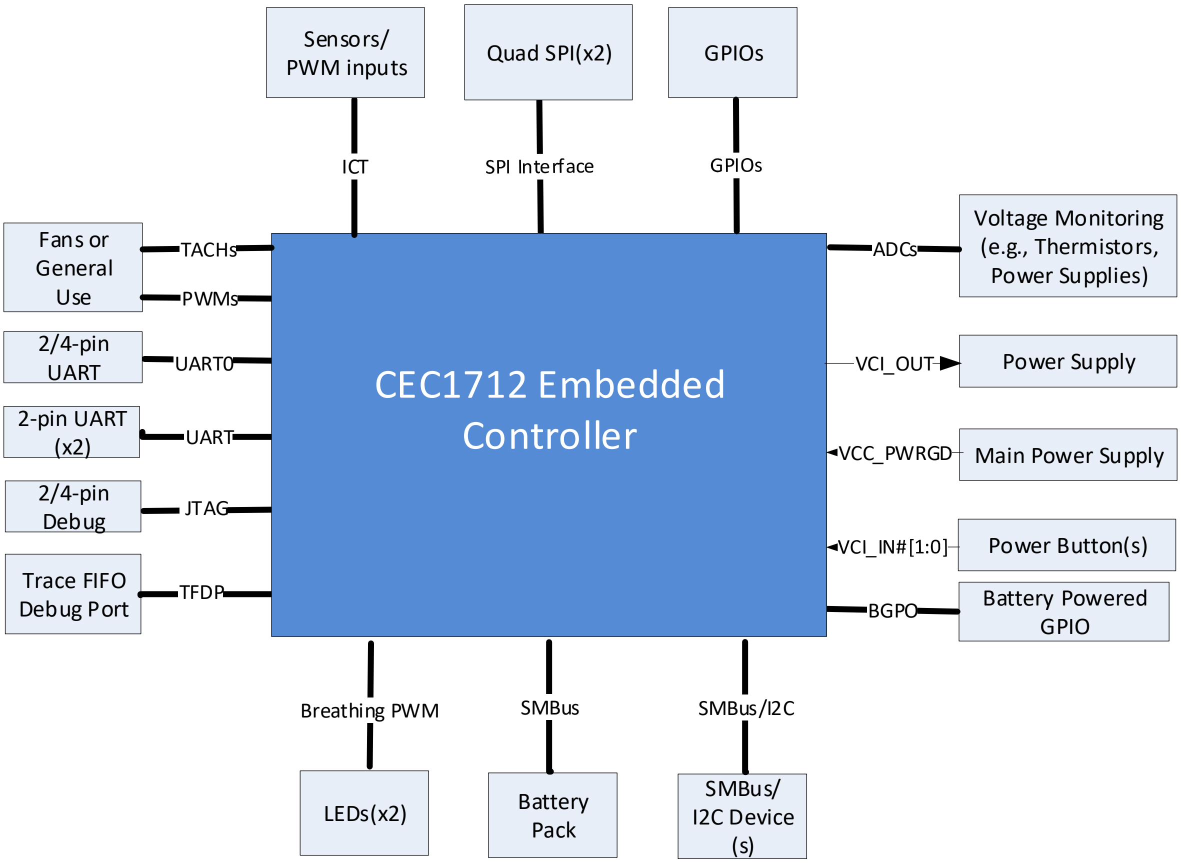
Note 1: Please refer to Boot ROM document for below set of optional OTP selectable feature.

1.2 Boot ROM

Following the release of the RESET_EC signal, the processor will start executing code in the Boot ROM. The Boot ROM executes the SPI Flash Loader, which downloads User Code from SPI Flash and stores it in the internal Code RAM. Refer to CEC1712 Boot ROM document for further details.

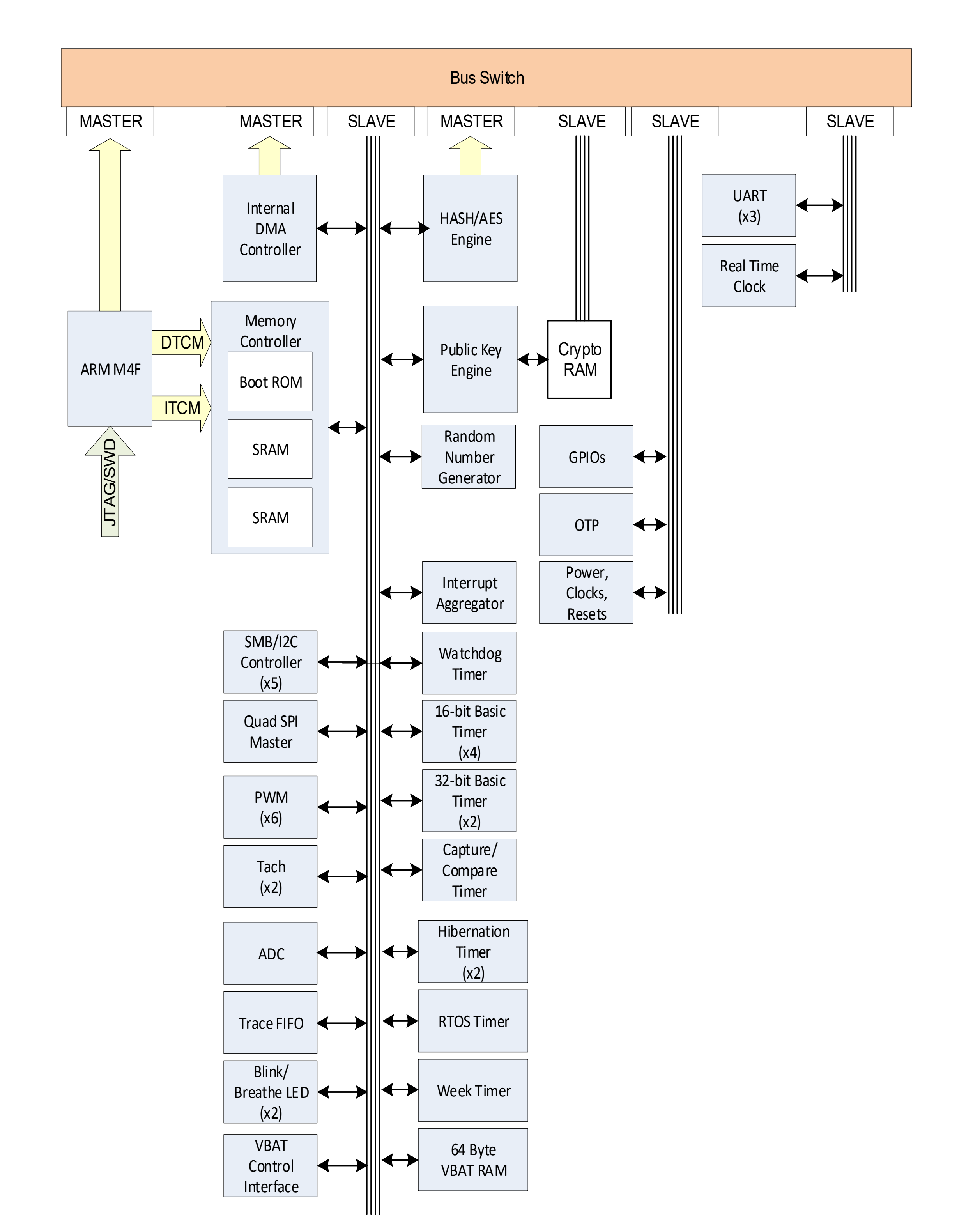
1.3 System Block Diagram

1.3.1 CEC1712 INTERNAL ADDRESS SPACES



The Internal Embedded Controller can access any register in the EC Address Space or Host Address Space.

FIGURE 1-1: BLOCK DIAGRAM



2.0 PIN CONFIGURATION

2.1 Description

The Pin Configuration chapter includes Pin List, Pin Multiplexing.

2.2 Terminology and Symbols for Pins/Buffers

2.2.1 BUFFER TERMINOLOGY

Term	Definition
#	The '#' sign at the end of a signal name indicates an active-low signal
n	The lowercase 'n' preceding a signal name indicates an active-low signal
PWR	Power
PIO	Programmable as Input, Output, Open Drain Output, Bi-directional or Bi-directional with Open Drain Output. Configurable drive strength from 2ma to12ma. **Note:** All GPIOs have programmable drive strength options of 2ma, 4ma, 8ma and 12ma. GPIO pin drive strength is determined by the Pin Control Register Defaults field in the Pin Control Register 2.
In	I Type Input Buffer.
O2	O-2 mA Type Buffer.
PECI	PECI Input/Output. These pins operate at the processor voltage level (VREF_VTT)
SB-TSI	SB-TSI Input/Output. These pins operate at the processor voltage level (VREF_VTT)

2.2.2 PIN NAMING CONVENTIONS

- Pin Name is composed of the multiplexed options separated by '/'. E.g., GPIOxxxx/SignalA/SignalB.
- The first signal shown in a pin name is the default signal. E.g., GPIOxxxx/SignalA/SignalB means the GPIO is the default signal.
- Parenthesis '()' are used to list aliases or alternate functionality for a single mux option.
- Square brackets '[ ]' are used to indicate there is a Strap Option on a pin. This is always shown as the last signal on the Pin Name.
- Signal Names appended with a numeric value indicates the Instance Number. E.g., PWM0, PWM1, etc. indicates that PWM0 is the PWM output for PWM Instance 0, PWM1 is the PWM output for PWM Instance 1, etc. The instance number may be omitted if there in only one instance of the IP block implemented.

2.3 Pin List

TABLE 2-1: CEC1712 PIN MAP

Ball map	Signal
D2	nRESET_IN
F3	GPIO057/VCC_PWRGD
E2	GPIO106/PWROK
B2	GPIO051/ICT1_TACH1
A3	GPIO050/ICT0_TACH0
F2	GPIO200/ADC00/TRACEDAT0
G2	GPIO201/ADC01/TRACEDAT1
H2	GPIO202/ADC02/TRACEDAT2
G1	GPIO203/ADC03/TRACEDAT3
H1	GPIO204/ADC04

Ball map	Signal
D1	VSS
J8	GPIO070/I2C14_SDA
K8	GPIO071/I2C14_SCL
J6	GPIO063/PWM6_ALT/ICT8
K3	GPIO224/SHD_IO1
K2	GPIO016/SHD_IO3/ICT3
K4	GPIO227/SHD_IO2
K5	GPIO223/SHD_IO0
K7	GPIO055/PWM2/SHD_CS0#
K6	GPIO056/PWM3/SHD_CLK
K1	GPIO012/I2C07_SDA
J2	GPIO013/I2C07_SCL
J5	GPIO130/I2C01_SDA
J9	GPIO131/I2C01_SCL
J3	GPIO020
J4	GPIO021
J7	GPIO002/PWM5/SHD_CS1#
H6	GPIO015/PWM7/ICT10
H5	GPIO032
A9	GPIO132/I2C06_SDA
B7	GPIO140/I2C06_SCL/ICT5
K9	GPIO026/I2C12_SDA
K10	GPIO053/PWM0
J10	GPIO027/I2C12_SCL
G7	GPIO030/I2C10_SDA
H9	GPIO107/I2C10_SCL
H10	GPIO120
G9	GPIO112
G10	GPIO113/ICT9
G4	GPIO034
F9	GPIO170/UART1_TX[JTAG_STRAP]
F8	GPIO171/UART1_RX
E8	JTAG_RST#
D9	GPIO104/UART0_TX/TFDP_CLK[VTR2_STRAP]
E9	GPIO105/UART0_RX/TFDP_DATA/TRACECLK
C9	GPIO046/ICT11
B8	GPIO047/PWM3_ALT/ICT13
D10	GPIO121/PVT_IO0
B10	GPIO122/PVT_IO1
E10	GPIO123/PVT_IO2
F10	GPIO126/PVT_IO3
C10	GPIO124/PVT_CS#/ICT12
A10	GPIO125/PVT_CLK
C6	GPIO127

Ball map	Signal
D7	GPIO156/LED0
B9	GPIO157/LED1
C5	GPIO045/PWM2_ALT/ICT14
A6	GPIO165/32KHZ_IN/CTOUT0
C2	GPIO145/I2C09_SDA/JTAG_TDI/UART2_RX
B6	GPIO146/I2C09_SCL/JTAG_TDO/UART2_TX
A7	GPIO147/I2C15_SDA/JTAG_CLK
B3	GPIO150/I2C15_SCL/JTAG_TMS
E7	GPIO143/I2C04_SDA/UART0_CTS#
D6	GPIO144/I2C04_SCL/UART0_RTS#
A8	GPIO004/I2C00_SCL
B5	GPIO003/I2C00_SDA
A5	VCI_IN3#/GPIO000
B4	VCI_IN0#/GPIO163
B1	BGPO0/GPIO253
A1	VCI_OUT/GPIO250
A4	XTAL1
A2	XTAL2
D4	VSS_ANALOG
C1	VTR_PLL
D5	VBAT
E4	VSS
E1	VTR_REG
E3	VREF_ADC
F7	VSS
G6	VTR1
F4	VTR_ANALOG
F1	VR_CAP
G5	VTR2
J1	VSS_ADC

Note: GPIO055/PWM2/SHD_CS0# should be pulled up for proper boot up of the chip.

2.4 Pin Multiplexing

2.4.1 DEFAULT STATE

The default state for analog pins is Input. The default state for all pins that default to a GPIO function is input/output/interrupt disabled. The default state for pins that differ is shown in the Section 3.5, "GPIO Register Assignments". Entries for the Default State column are

- O2ma-Low: Push-Pull output, Slow slew rate, 2ma drive strength, grounded
- O2ma-High Push-Pull output, Slow slew rate, 2ma drive strength, high output
- PU Input, with pull-up resistor enabled

2.4.2 POWER RAIL

The Power Rail column defines the power pin that provides I/O power for the signal pin.

2.4.3 BUFFER TYPES

The Buffer Type column defines the type of Buffer associated with each signal. Some pins have signals with two different buffer types sharing the pin; in this case, table shows the buffer type for each of the signals that share the pin.

Input signals muxed with GPIOs are marked as “I”

Output signals muxed with GPIOs are marked as “PIO”, because the GPIO input path is always active even when the alternate function selected is “output only”. So the GPIO input can be read to see the level of the output signal.

Pad Types are defined in the Section 33.0, "Electrical Specifications," on page 308.

- I/O Pad Types are defined in Section 33.2.4, "DC Electrical Characteristics for I/O Buffers," on page 310.
- The abbreviation “PWR” is used to denote power pins. The power supplies are defined in Section 33.2.1, "Power Supply Operational Characteristics," on page 308.

2.4.4 GLITCH PROTECTION

Pins with glitch protection are glitch-free tristate pins and will not drive out while their associated power rail is rising. These glitch-free tristate pins require either an external pull-up or pull-down to set the state of the pin high or low.

Note: If the pin needs to default low, a 1M ohm (max) external pull-down is required.

All pins are glitch protected.

Note: The power rail must rise monotonically in order for glitch protection to operate.

2.4.5 OVER-VOLTAGE PROTECTION

If a pin is over-voltage protected (over-voltage protection = YES) then the following is true: If the pad is powered by 1.8V +/- 5% (operational) it can tolerate up to 3.63V on the pad. This allows for a pull-up to 3.3V power rail +/- 10%. If the pad is powered by 3.3V +/- 5% (operational) it can tolerate up to 5.5V on the pad. This allows for a pull-up to 5.0V power rail +/- 10%.

If a pin is not over-voltage protected (over-voltage protection = NO) then the following is true: If the pad is powered by 1.8V +/- 5% (operational), it can tolerate up to 1.8V +10% (i.e., +1.98V max). If the pad is powered by 3.3V +/- 5% (operational) it can tolerate up to 3.3V +10% (i.e., +3.63V max).

2.4.6 UNDER-VOLTAGE PROTECTION

Pins that are identified as having Under-voltage PROTECTION may be configured so they will not sink excess current if powered by 3.3V and externally pulled up to 1.8V. The following configuration requirements must be met.

- If the pad is an output only pad type and it is configured as either open drain or the output is disabled.
- If the pin is a GPIO pin with a PIO pad type then is must be configured as open drain output with the input disabled. The input is disabled by setting the GPIO Power Gating Signals (PGS) bits to 11b.

All pins are under voltage protected.

2.4.7 BACKDRIVE PROTECTION

Assuming that the external voltage on the pin is within the parameters defined for the specific pad type, the backdrive protected pin will not sink excess current when it is at a lower potential than the external circuit. There are two cases where this occurs:

- The pad power is off and the external circuit is powered
- The pad power is on and the external circuitry is pulled to a higher potential than the pad power. This may occur on 3.3V powered pads that are 5V tolerant or on 1.8V powered pads that are 3.6V tolerant.

2.4.8 EMULATED POWER WELL

Power well emulation for GPIOs and for signals that are multiplexed with GPIO signals is controlled by the Power Gating Signals (PGS) option in the GPIO Pin Control Register. The Emulated Power Well column in the Pin Multiplexing table defines the power gating programming options supported for each signal.

Note: VBAT powered signals do not support power emulation and must program the PGS bit field to 00b (VTR)

2.4.9 GATED STATE

This column defines the internal value of an input signal when either its emulated power well is inactive or it is not selected by the GPIO alternate function MUX. A value of “No Gate” means that the internal signal always follows the pin even when the emulated power well is inactive.

Note: Gated state is only meaningful to the operation of input signals. A gated state on an output pin defines the internal behavior of the GPIO MUX and does not imply pin behavior.

Note: Only the pins that are 5V tolerant have an entry in the 5VT column in the Pin Description Table.

2.4.10 NOTES

The below notes are for all tables in this chapter.

TABLE 2-2: NUMBERED NOTES

NOTE	Description
Note 1	An external cap must be connected as close to the VR_CAP pin/ball as possible with a routing resistance and CAP ESR of less than 100mohms. The capacitor value is 1uF and must be ceramic with X5R or X7R dielectric. The cap pin/ball should remain on the top layer of the PCB and traced to the CAP. Avoid adding vias to other layers to minimize inductance.
Note 2	This SMBus ports supports 1 Mbps operation as defined by I2C. For 1 Mbps I2C recommended capacitance/pull-up relationships from Intel, refer to the Shark Bay platform guide, Intel ref number 486714. Refer to the PCH - SMBus 2.0/SMLink Interface Design Guidelines, Bus Capacitance/Pull-Up Resistor Relationship.
Note 4	The voltage on the ADC pins must not exceed 3.6 V or damage to the device will occur.
Note 5	The VCI pins may be used as GPIOs. The VCI input signals are not gated by selecting the GPIO alternate function. Firmware must disable (i.e., gate) these inputs by writing the bits in the VCI Input Enable Register when the GPIO function is enabled.
Note 6	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration register
Note 7	Refer Configurable Signal Routing section under Pin Configuration chapter for details on using the <signal> and <signal>_ALT. Both <signal> and <signal>_ALT cannot be enabled simultaneously.
Note 8	<Signal> with ‘#’ as suffix will be shown as <Signal>_n in MPLab Tools
Note 9	32kHz_IN is named CLK32kHz_IN in MPLab Tools
Note 10	Clock Enable Register Bits [3:2] should be configured to be driven by single ended 32Khz source. Connect the pin to SUSCLK from PCH.
Note 11	When the JTAG_RST# pin is not asserted (logic'1'), the JTAG or ARM SWJ signal functions in the JTAG interface are unconditionally routed to the GPIO interface; the Pin Control register for these GPIO pins has no effect.When the JTAG_RST# pin is asserted (logic'0'), the signal functions in the JTAG interface are not routed to the interface and the Pin Control Register for these GPIO pins controls the muxing. The pin control registers can not route the JTAG interface to the pins. System Board Designer should terminate this pin in all functional state using jumpers and pull-up or pull down resistors, etc.
Note 12	The JTAG signals TDI,TDO,TMS,TCK are muxed with GPIO pins. Routing of JTAG signals to these pins are dependent on DEBUG ENABLE REGISTER bits [2:0] and JTAG_RST# pin (Note . To configure these GPIO pins for non JTAG functions, pull JTAG_RST# low externally and select the appropriate alternate function in the Pin Control Register
Note 13	The BGPO pins may be used as GPIO. For this the BGPO power control register and GPIO pin control register needs to be configured
Note 14	GPIO000/VCI_IN3#, if not used must be connected to VBAT through a high impedance resistor of the order of 100k

TABLE 2-2: NUMBERED NOTES

NOTE	Description
Note 15	External pull up should be added on GPIO055/SHD_CS0# pin for proper booting

2.4.11 CEC1712 MULTIPLEXING

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	nRESET_IN	I		VTR1	PGS=00 (only)			Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO057	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	VCC_PWRGD	PIO			PGS=00 (only)	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO106	PIO		VTR1	All PGS options	No Gate		Yes	
1	PWROK	PIO			PGS=00 (only)	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO051	PIO		VTR1	All PGS options	No Gate		Yes	
1	ICT1_TACH1	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO050	PIO		VTR1	All PGS options	No Gate		Yes	
1	ICT0_TACH0	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO200	PIO		VTR1	All PGS options	No Gate		No	

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVoltage Protect	Back-drive Protect	Notes
1	ADC00	I_AN			PGS=00 (only)	Low			Note 4
2	Reserved								
3	Reserved								
Default: 0	GPIO201	PIO		VTR1	All PGS options	No Gate		No	
1	ADC01	I_AN			PGS=00 (only)	Low			Note 4
2	Reserved								
3	Reserved								
Default: 0	GPIO202	PIO		VTR1	All PGS options	No Gate		No	
1	ADC02	I_AN			PGS=00 (only)	Low			Note 4
2	Reserved								
3	Reserved								
Default: 0	GPIO203	PIO		VTR1	All PGS options	No Gate		No	
1	ADC03	I_AN			PGS=00 (only)	Low			Note 4
2	Reserved								
3	Reserved								
Default: 0	GPIO204	PIO		VTR1	All PGS options	No Gate		No	
1	ADC04	I_AN			PGS=00 (only)	Low			Note 4
2	Reserved								
3	Reserved								
Default: 0	GPIO070	PIO		VTR2	All PGS options	No Gate		Yes	
1	Reserved								
2	I2C14_SDA	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO071	PIO		VTR2	All PGS options	No Gate		Yes	
1	Reserved								
2	I2C14_SCL	PIO			All PGS options	High			
3	Reserved								

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO063	PIO		VTR2	All PGS options	No Gate		Yes	
1	Reserved								
2	PWM6_ALT	PIO			All PGS options	NA			Note 7
3	ICT8	I			All PGS options	Low			
Default: 0	GPIO224	PIO		VTR2	All PGS options	No Gate		Yes	
1	Reserved								
2	SHD_IO1	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO016	PIO		VTR2	All PGS options	No Gate		Yes	
1	Reserved								
2	SHD_IO3	PIO			All PGS options	Low			
3	ICT3	PIO			All PGS options	Low			
Default: 0	GPIO227	PIO		VTR2	All PGS options	No Gate		Yes	
1	SHD_IO2	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO223	PIO		VTR2	All PGS options	No Gate		Yes	
1	SHD_IO0	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO055	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM2	PIO			All PGS options	NA			
2	SHD_CS0#	PIO			All PGS options	NA			Note 15
3	Reserved								

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO056	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM3	PIO			All PGS options	NA			
2	SHD_CLK	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO012	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C07_SDA	PIO			All PGS options	High			Note 2
2	Reserved								
3	Reserved								
Default: 0	GPIO013	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C07_SCL	PIO			All PGS options	High			Note 2
2	Reserved								
3	Reserved								
Default: 0	GPIO130	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C01_SDA	PIO			All PGS options	High			Note 2
2	Reserved								
3	Reserved								
Default: 0	GPIO131	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C01_SCL	PIO			All PGS options	High			Note 2
2	Reserved								
3	Reserved								
Default: 0	GPIO020	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO021	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	Reserved								
Default: 0	GPIO002	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM5	PIO			All PGS options	NA			
2	SHD_CS1#	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO015	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM7	PIO			All PGS options	NA			
2	ICT10	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO032	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO132	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C06_SDA	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO140	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C06_SCL	PIO			All PGS options	High			
2	ICT5	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO026	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	I2C12_SDA	PIO			All PGS options	High			

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVoltage Protect	Back-drive Protect	Notes
Default: 0	GPIO053	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM0	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO027	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	I2C12_SCL	PIO			All PGS options	High			
Default: 0	GPIO030	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	I2C10_SDA	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO107	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	I2C10_SCL	PIO			All PGS options	High			
Default: 0	GPIO120	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO112	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO113	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	ICT9	I			All PGS options	Low			

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	Reserved								
Default: 0	GPIO034	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO170	PIO	PU	VTR1	All PGS options	No Gate		Yes	
1	UART1_TX	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Strap	JTAG_STRAP	PIO							
Default: 0	GPIO171	PIO		VTR1	All PGS options	No Gate		Yes	
1	UART1_RX	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	JTAG_RST#	I		VTR1	N/A			Yes	Note 11,12
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO104	PIO		VTR1	All PGS options	No Gate		Yes	
1	UART0_TX	PIO			All PGS options	NA			
2	TFDP_CLK	PIO			All PGS options	NA			
3	Reserved								
Strap	VTR2_STRAP	PIO							
Default: 0	GPIO105	PIO		VTR1	All PGS options	No Gate		Yes	
1	UART0_RX	PIO			All PGS options	Low			
2	TFDP_DATA	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO046	PIO		VTR1	All PGS options	No Gate		Yes	

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
1	KSO2	PIO			All PGS options	NA			
2	Reserved								
3	ICT11	I			All PGS options	Low			
Default: 0	GPIO047	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	PWM3_ALT	PIO			All PGS options	NA			Note 7
3	ICT13	I			All PGS options	Low			
Default: 0	GPIO121	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_IO0	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO122	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_IO1	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO123	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	PVT_IO2	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO126	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_IO3	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO124	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_CS#	PIO			All PGS options	NA			

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
2	Reserved								
3	ICT12	I			All PGS options	Low			
Default: 0	GPIO125	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_CLK	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO127	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO156	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	LED0	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO157	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	LED1	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO045	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	PWM2_ALT	PIO			All PGS options	NA			
3	ICT14	I			All PGS options	Low			
Default: 0	GPIO165	PIO		VTR1	All PGS options	No Gate		Yes	
1	32KHZ_IN	PIO			PGS=00 (only)	Low			
2	Reserved								
3	CTOUT0	PIO			All PGS options	NA			

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO145	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C09_SDA	PIO			All PGS options	High			
2	UART2_RX	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO146	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C09_SCL	PIO			All PGS options	High			
2	UART2_TX	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO147	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C15_SDA	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO150	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C15_SCL	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO143	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C04_SDA	PIO			All PGS options	High			
2	UART0_CTS#	I			All PGS options	High			
3	Reserved								
Default: 0	GPIO144	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C04_SCL	PIO			All PGS options	High			
2	UART0_RTS#	PIO			All PGS options	NA			
3	Reserved								

TABLE 2-3: CEC1712 PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO004	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C00_SCL	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO003	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C00_SDA	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
0	GPIO000	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_IN3#	ILLK			PGS=00 (only)	No Gate			Note 14
2	Reserved								
3	Reserved								
0	GPIO163	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_IN0#	ILLK			PGS=00 (only)	No Gate			Note 5
2	Reserved								
3	Reserved								
0	GPIO253	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	BGPO0	PIO	O2ma-Low		PGS=00 (only)	NA			Note 13
2	Reserved								
3	Reserved								
0	GPIO250	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_OUT	PIO	O2ma-High		PGS=00 (only)	NA			
2	Reserved								
3	Reserved								

2.5 Configurable Signal Routing

To accommodate the signal routing across packages, some Signals are routed to more than one GPIO. At any given time, only the <Signal> or <Signal>_ALT can be selected. Both cannot be selected at the same time.

2.5.1 SIGNAL DESCRIPTION BY INTERFACE

TABLE 2-4: SIGNAL DESCRIPTION BY INTERFACE

SIG_NAME	Description	Notes
ADC		
ADCxx	ADC channel input	Note 5 'xx' is the index of the ADC input. Refer Family features table to find the number of ADC inputs supported in the package
Miscellaneous		
I2C/SMBus Controller		
I2Cxx_SDA	I2C/SMBus Controller Port 0 Data	Note 2 'xx' is the index of the I2C port. Refer Family features table to find the number of I2C ports supported in the package
I2Cxx_SCL	I2C/SMBus Controller Port 0 Clock	Note 2
GPIO		
GPIOx	General Purpose Input Output Pins	
PCR Interface		
32KHZ_OUT	32.768 KHz Digital Output	
32KHZ_IN	32.768 KHz Digital Input	
nRESET_IN	External System Reset Input	
PECI		
PECI_DAT	PECI Bus	
VREF_VTT	Processor Interface Voltage Reference	
Quad Mode SPI Controller ports		
PVT_CS#	Private SPI Chip Select	SPI_CS0# of QMSPI Controller
PVT_IO0	Private SPI Data 0	SPI_IO0 of QMSPI Controller
PVT_IO1	Private SPI Data 1	SPI_IO1 of QMSPI Controller
PVT_IO2	Private SPI Data 2	SPI_IO2 of QMSPI Controller
PVT_IO3	Private SPI Data 3	SPI_IO3 of QMSPI Controller
PVT_CLK	Private SPI Clock	SPI_CLK of QMSPI Controller
SHD_CS1#	Shared SPI Chip Select1	SPI_CS1# of QMSPI Controlelr
SHD_CS0#	Shared SPI Chip Select	SPI_CS0# of QMSPI Controller
SHD_IO0	Shared SPI Data 0	SPI_IO0 of QMSPI Controller
SHD_IO1	Shared SPI Data 1	SPI_IO1 of QMSPI Controller
SHD_IO2	Shared SPI Data 2	SPI_IO2 of QMSPI Controller
SHD_IO3	Shared SPI Data 3	SPI_IO3 of QMSPI Controller
SHD_CLK	Shared SPI Clock	SPI_CLK of QMSPI Controller

TABLE 2-4: SIGNAL DESCRIPTION BY INTERFACE (CONTINUED)

SIG_NAME	Description	Notes
FAN PWM and Tachometer		
ICT0_TACH0	Fan Tachometer Input 0	
ICT1_TACH1	Fan Tachometer Input 1	
ICT2_TACH2	Fan Tachometer Input 2	
TACH3	Fan Tachometer Input 3	
PWMx	Pulse Width Modulator Output	'x' is the index of the PWM output. Refer Family features table to find the number of PWM outputs supported in the package
Input Capture/Compare timer		
ICTx	Input capture timer input	'x' is the index of the PWM output. Refer Family features table to find the number of ICT inputs supported in the package
CTOUT0	Compare timer 0 toggle output	
CTOUT1	Compare timer 1 toggle output	
Serial ports		
UART_CLK	UART Baud Clock Input	
UART0_RX	UART Receive Data (RXD)	
UART0_TX	UART Transmit Data (TXD)	
UART0_CTS#	Clear to Send Input	
UART0_RTS#	Request to Send Output	
UART0_RI#	Ring Indicator Input	
UART0_DCD#	Data Carrier Detect Input	
UART0_DSR#	Data Set Ready Input	
UART0_DTR#	Data Terminal Ready Output	
JTAG		
JTAG_RST#	JTAG test active low reset	Note 11,12
JTAG_TDI	JTAG test data in	Note 11,12
JTAG_TDO	JTAG test data out	Note 11,12
JTAG_CLK	JTAG test clk; SWDCLK	Note 11,12
JTAG_TMS	JTAG test mode select; SWDIO	Note 11,12
TFDP_DATA	Trace FIFO debug port - data	
TFDP_CLK	Trace FIFO debug port - clock	
TRACECLK	ARM Embedded Trace Macro Clock	Trace Port is enabled by setting TRACE_EN bit of ETM Trace enable register in EC Register Bank
TRACEDATA0	ARM Embedded Trace Macro Data 0	
TRACEDATA1	ARM Embedded Trace Macro Data 1	
TRACEDATA2	ARM Embedded Trace Macro Data 2	
TRACEDATA3	ARM Embedded Trace Macro Data 3	
Power pins		
VREF_ADC	ADC Reference Voltage	
VSS_ADC	Analog ADC supply associated ground	
VBAT	VBAT supply	
VR_CAP	Internal Voltage Regulator Capacitor	Note 1

TABLE 2-4: SIGNAL DESCRIPTION BY INTERFACE (CONTINUED)

SIG_NAME	Description	Notes
VSS	VTR associated ground	
VSS_VBAT	VBAT associated ground	
VTR1	VTR Suspend Power Supply	
VTR2	Peripheral Power Supply	
VTR_PLL	PLL power supply	
VTR_REG	Main Regulator Power supply	

2.5.2 STRAPPING OPTIONS

GPIO170 is used for the TAP Controller select strap. If any of the JTAG TAP controllers are used, GPIO170 must only be configured as an output to a VTRx powered external function. GPIO170 may only be configured as an input when the JTAG TAP controllers are not needed or when an external driver does not violate the Slave Select Timing.See Section 32.2.1, "TAP Controller Select Strap Option".

TABLE 2-5: STRAP PINS

Pin Name	Strap Name	Strap Define and Value	I/O Power Rail
GPIO170	JTAG_STRAP	1= Boundary Scan The JTAG Port is used to access the Boundary scan TAP controller 0= Normal Operation The JTAG port is used to access the ARM TAP Controller	VTR1
GPIO104	VTR2_STRAP	Voltage Level strap is used to determine if the Shared Flash interface must be configured for 3.3V or 1.8V operation 1= 3.3V Operation 0= 1.8V Operation	VTR1

2.6 Pin Default State Through Power Transitions

The power state and power state transitions illustrated in the following tables are defined in Section 4.0, "Power, Clocks, and Resets". Pin behavior in this table assumes no specific programming to change the pin state. All GPIO default pins that have the same behavior are described in the table generically as GPIOXXX.

TABLE 2-6: PIN DEFAULT STATE THROUGH POWER TRANSITIONS

Signal	VBAT Applied	VBAT Stable	VTR Applied	RESET_SYS De-asserted	RESET_SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
GPIO170	un-powered	un-powered	High	In	Z	glitch	un-powered	
GPIOXXX	un-powered	un-powered	Z	Z	Z	glitch	un-powered	Note D
nRESET_IN	un-powered	un-powered	Low	In	Z	glitch	un-powered	
BGPOx	Out=0	Out=0	Retain	Retain	Retain	Retain	un-powered	Note B
VCI_INx#	In	In	In	In	In	In	un-powered	

TABLE 2-6: PIN DEFAULT STATE THROUGH POWER TRANSITIONS

Signal	VBAT Applied	VBAT Stable	VTR Applied	RESET_SYS De-asserted	RESET_SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
VCI_OUT	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	un-powered	Note C
XTAL1	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	
XTAL2	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	

Legend	Notes
(P) = I/O state is driven by protocol while power is applied. **Z** = Tristate **In** = Input	Note D: Does not include GPIO170 Note B: Pin is programmable by the EC and retains its value through a VTR power cycle.

2.7 Package Information

2.7.1 84 PIN WFBGA/2ZW PACKAGE

FIGURE 2-1: 84-BALL VERY, VERY THIN FINE PITCH BALL GRID ARRAY (2ZW) - 7X7X0.8 MM BODY [WFBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

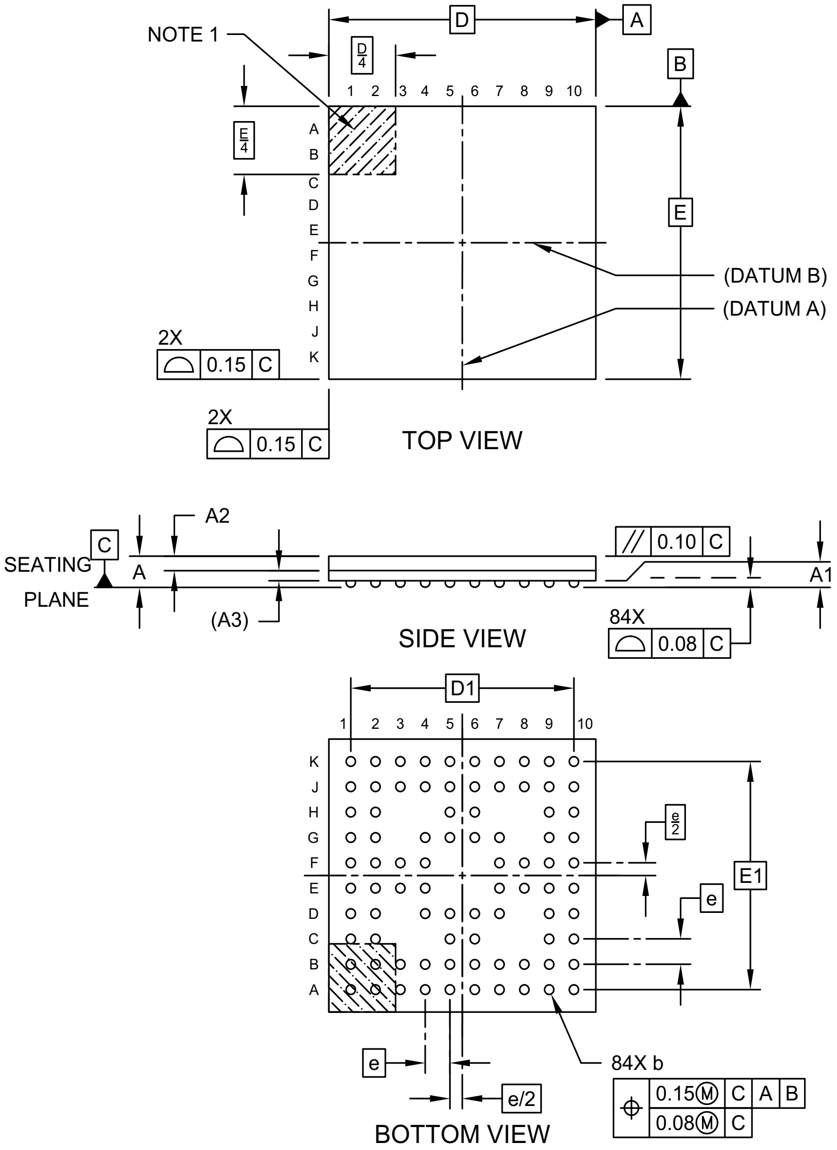
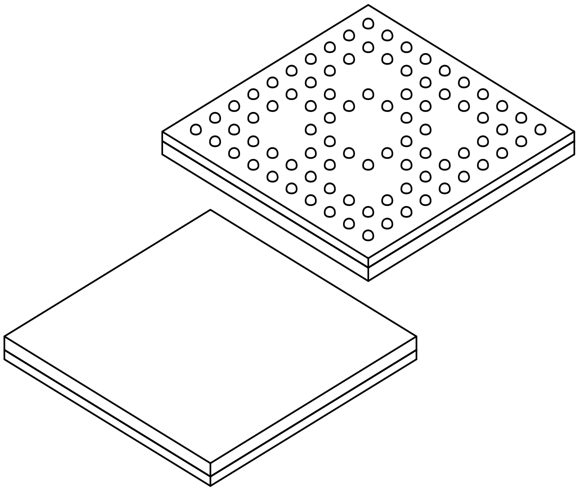


FIGURE 2-1: 84-BALL VERY, VERY THIN FINE PITCH BALL GRID ARRAY (2ZW) - 7X7X0.8 MM BODY [WFBGA] (CONTINUED)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



		Units	MILLIMETERS	
		Dimension Limits	MIN	NOM	MAX
Number of Terminals	N		84	
Pitch	e		0.65 BSC	
Overall Height	A	-	-	0.80
Standoff	A1	0.12	0.17	-
Mold Cap Thickness	A2	0.35	0.40	0.45
Substrate Thickness	A3		0.13 REF	
Overall Length	D		7.00 BSC	
Overall Termnal Spacing	D1		5.85 BSC	
Overall Width	E		7.00 BSC	
Overall Termnal Spacing	E1		5.85 BSC	
Ball Diameter	b	0.20	0.25	0.30

Notes:

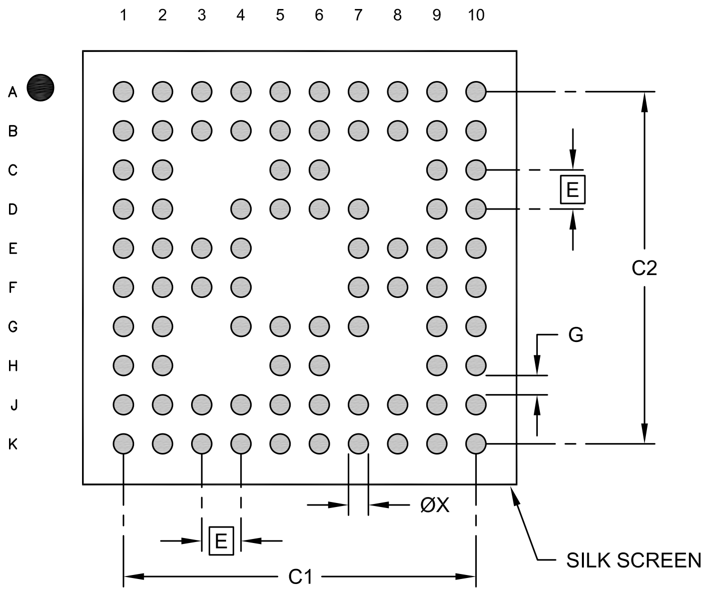
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

FIGURE 2-1: 84-BALL VERY, VERY THIN FINE PITCH BALL GRID ARRAY (2ZW) - 7X7X0.8 MM BODY [WFBGA] (CONTINUED)

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Overall Contact Pad Spacing	C1		5.85	
Overall Contact Pad Spacing	C2		5.85	
Contact Pad Width (X84)	X1			0.33
Contact Pad to Contact Pad	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

3.0 DEVICE INVENTORY

3.1 Conventions

Term	Definition
Block	Used to identify or describe the logic or IP Blocks implemented in the device.
Reserved	Reserved registers and bits defined in the following table are read only values that return 0 when read. Writes to these reserved registers have no effect.
TEST	Microchip Reserved locations which should not be modified from their default value. Changing a TEST register or a TEST field within a register may cause unwanted results.
b	The letter ‘b’ following a number denotes a binary number.
h	The letter ‘h’ following a number denotes a hexadecimal number.

Register access notation is in the form “Read / Write”. A Read term without a Write term means that the bit is read-only and writing has no effect. A Write term without a Read term means that the bit is write-only, and assumes that reading returns all zeros.

Register Field Type	Field Description
R	**Read:** A register or bit with this attribute can be read.
W	**Write:** A register or bit with this attribute can be written.
RS	**Read to Set:** This bit is set on read.
RC	**Read to Clear:** Content is cleared after the read. Writes have no effect.
WC or W1C	**Write One to Clear:** writing a one clears the value. Writing a zero has no effect.
WZC	**Write Zero to Clear:** writing a zero clears the value. Writing a one has no effect.
WS or W1S	**Write One to Set:** writing a one sets the value to 1. Writing a zero has no effect.
WZS	**Write Zero to Set:** writing a zero sets the value to 1. Writing a one has no effect.

3.2 Block Overview and Base Addresses

Table 3-1, "Base address" lists all the IP components, referred to as Blocks, implemented in the design. The registers implemented in each block are accessible by the embedded controller (EC) at an offset from the Base Address shown in Table 3-1, "Base address". The registers can also be accessed by various hosts in the system as below

1. I2C : I2C host access is handled by firmware
2. JTAG : JTAG port has access to all the registers defined in Table 3-1, "Base address".

TABLE 3-1: BASE ADDRESS

Feature	Instance	Logical Device Number	Base Address
Watchdog Timer			4000_0400h
16-bit Basic Timer	0		4000_0C00h
16-bit Basic Timer	1		4000_0C20h
32-bit Basic Timer	0		4000_0C80h
32-bit Basic Timer	1		4000_0CA0h
Capture-Compare Timers			4000_1000h
DMA Controller			4000_2400h
SMB-I2C Controller	0		4000_4000h
SMB-I2C Controller	1		4000_4400h
SMB-I2C Controller	2		4000_4800h
SMB-I2C Controller	3		4000_4C00h
SMB-I2C Controller	4		4000_5000h
I2C Controller	5		4000_5100h
I2C Controller	6		4000_5200h
I2C Controller	7		4000_5300h
Quad Master SPI			4007_0000h
16-bit PWM	0		4000_5800h
16-bit PWM	2		4000_5820h
16-bit PWM	3		4000_5830h
16-bit PWM	5		4000_5850h
16-bit PWM	6		4000_5860h
16-bit PWM	7		4000_5870h
16-bit Tach	0		4000_6000h
16-bit Tach	1		4000_6010h
RTOS Timer			4000_7400h
ADC			4000_7C00h
Trace FIFO			4000_8C00h
Hibernation Timer	0		4000_9800h
Hibernation Timer	1		4000_9820h
VBAT Register Bank			4000_A400h
VBAT Powered RAM			4000_A800h
Week Timer			4000_AC80h
VBAT-Powered Control Interface			4000_AE00h
Blinking-Breathing LED	0		4000_B800h
Blinking-Breathing LED	1		4000_B900h
Interrupt Aggregator			4000_E000h

TABLE 3-1: BASE ADDRESS

Feature	Instance	Logical Device Number	Base Address
EC Subsystem Registers			4000_FC00h
JTAG			4008_0000h
Power, Clocks and Resets			4008_0100h
GPIOs			4008_1000h
UART	0	9h	400F_2400h
UART	1	Ah	400F_2800h
UART	2	Bh	400F_2C00h
Real Time Clock		14h	400F_5000h
Global Configuration		3Fh	400F_FF00h

3.3 Sleep Enable Register Assignments

TABLE 3-2: SLEEP ALLOCATION

Block	Instance	Bit Position	Sleep Enable Register	Clock Required Register	Reset Enable Register
JTAG STAP		0	NA	Clock Required 0	NA
Interrupt		0	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	0	2	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	0	4	Sleep Enable 1	Clock Required 1	Reset Enable 1
DMA		6	Sleep Enable 1	Clock Required 1	Reset Enable 1
TFDP		7	Sleep Enable 1	Clock Required 1	Reset Enable 1
PROCESSOR		8	Sleep Enable 1	Clock Required 1	NA
WDT		9	NA	Clock Required 1	Reset Enable 1
SMB	0	10	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	1	11	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	2	21	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	3	22	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	5	24	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	6	25	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	7	26	Sleep Enable 1	Clock Required 1	Reset Enable 1
EC Register Bank		29	Sleep Enable 1	Clock Required 1	NA
Basic Timer 16	0	30	Sleep Enable 1	Clock Required 1	Reset Enable 1
Basic Timer 16	1	31	Sleep Enable 1	Clock Required 1	Reset Enable 1
UART	0	1	Sleep Enable 2	Clock Required 2	Reset Enable 2
UART	1	2	Sleep Enable 2	Clock Required 2	Reset Enable 2
Global Configuration		12	NA	Clock Required 2	NA
RTC		18	NA	Clock Required 2	NA
ADC		3	Sleep Enable 3	Clock Required 3	Reset Enable 3
Hibernation Timer	0	10	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	1	13	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	2	14	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	3	15	Sleep Enable 3	Clock Required 3	Reset Enable 3
LED	0	16	Sleep Enable 3	Clock Required 3	Reset Enable 3
LED	1	17	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	4	20	Sleep Enable 3	Clock Required 3	Reset Enable 3
Basic Timer 32	0	23	Sleep Enable 3	Clock Required 3	Reset Enable 3
Basic Timer 32	1	24	Sleep Enable 3	Clock Required 3	Reset Enable 3
Hibernation Timer	1	29	Sleep Enable 3	Clock Required 3	Reset Enable 3
CCT	0	30	Sleep Enable 3	Clock Required 3	Reset Enable 3
RTOS Timer		6	NA	Clock Required 4	Reset Enable 4
Quad SPI Master		8	Sleep Enable 4	Clock Required 4	Reset Enable 4
I2C	5	10	Sleep Enable 4	Clock Required 4	Reset Enable 4
I2C	6	11	Sleep Enable 4	Clock Required 4	Reset Enable 4
I2C	7	12	Sleep Enable 4	Clock Required 4	Reset Enable 4

3.4 Interrupt Aggregator Bit Assignments

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
GIRQ8	0	GPIO140	GPIO Event	Yes	GPIO Interrupt Event	0	N/A
	1-2	Reserved					
	3	GPIO143	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO144	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO145	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO146	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO147	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO150	GPIO Event	Yes	GPIO Interrupt Event		
	9-13	Reserved					
	14	GPIO156	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO157	GPIO Event	Yes	GPIO Interrupt Event		
	16	Reserved					
	17	GPIO161	GPIO Event	Yes	GPIO Interrupt Event		
	18	Reserved					
	19	GPIO163	GPIO Event	Yes	GPIO Interrupt Event		
	20	Reserved					
	21	GPIO165	GPIO Event	Yes	GPIO Interrupt Event		
	22-23	Reserved					
	24	GPIO170	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO171	GPIO Event	Yes	GPIO Interrupt Event		
	25	Reserved					
	26	GPIO172	GPIO Event	Yes	GPIO Interrupt Event		
	27-28	Reserved					
	29	GPIO175	GPIO Event	Yes	GPIO Interrupt Event		
	30-31	Reserved					
GIRQ9	0-3	Reserved				1	N/A
	4	GPIO104	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO105	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO106	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO107	GPIO Event	Yes	GPIO Interrupt Event		
	8-9	Reserved					
	10	GPIO112	GPIO Event	Yes	GPIO Interrupt Event		
	11-15	Reserved					
	16	GPIO120	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO121	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO122	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO123	GPIO Event	Yes	GPIO Interrupt Event		

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	20	GPIO124	GPIO Event	Yes	GPIO Interrupt Event		
	20	Reserved					
	21	GPIO125	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO126	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO127	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO130	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO131	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO132	GPIO Event	Yes	GPIO Interrupt Event		
	27-31	Reserved					
GIRQ10	0-4	Reserved				2	N/A
	5	GPIO045	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO046	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO047	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO050	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO051	GPIO Event	Yes	GPIO Interrupt Event		
	10	Reserved					
	11	GPIO053	GPIO Event	Yes	GPIO Interrupt Event		
	12	Reserved					
	13	GPIO055	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO056	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO057	GPIO Event	Yes	GPIO Interrupt Event		
	16-18	Reserved					
	19	GPIO063	GPIO Event	Yes	GPIO Interrupt Event		
	20-23	Reserved					
	24	GPIO070	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO071	GPIO Event	Yes	GPIO Interrupt Event		
	26-31	Reserved					
GIRQ11	0	GPIO000	GPIO Event	Yes	GPIO Interrupt Event	3	N/A
	1	Reserved					
	2	GPIO002	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO003	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO004	GPIO Event	Yes	GPIO Interrupt Event		
	5-9	Reserved					
	10	GPIO012	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO013	GPIO Event	Yes	GPIO Interrupt Event		
	12	Reserved					
	13	GPIO015	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO016	GPIO Event	Yes	GPIO Interrupt Event		
	15-17	Reserved					

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	16	GPIO020	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO021	GPIO Event	Yes	GPIO Interrupt Event		
	18-21	Reserved					
	22	GPIO026	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO027	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO030	GPIO Event	Yes	GPIO Interrupt Event		
	25	Reserved					
	26	GPIO032	GPIO Event	Yes	GPIO Interrupt Event		
	27	Reserved					
	28	GPIO034	GPIO Event	Yes	GPIO Interrupt Event		
	29-31	Reserved					
GIRQ12	0	GPIO200	GPIO Event	Yes	GPIO Interrupt Event	4	N/A
	1	GPIO201	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO202	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO203	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO204	GPIO Event	Yes	GPIO Interrupt Event		
	5-18	Reserved					
	19	GPIO223	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO224	GPIO Event	Yes	GPIO Interrupt Event		
	21-22	Reserved					
	23	GPIO227	GPIO Event	Yes	GPIO Interrupt Event		
	24-31	Reserved					
GIRQ13	0	SMB-I2C Controller0	SMB-I2C	No	SMB-I2C Controller 0 Interrupt Event	5	20
	1	SMB-I2C Controller1	SMB-I2C	No	SMB-I2C Controller 1 Interrupt Event		21
	2	SMB-I2C Controller2	SMB-I2C	No	SMB-I2C Controller 2 Interrupt Event		22
	3	SMB-I2C Controller3	SMB-I2C	No	SMB-I2C Controller 3 Interrupt Event		23
	4	SMB-I2C Controller4	SMB-I2C	No	SMB-I2C Controller 4 Interrupt Event		158
	5	I2C Controller5	I2C	No	Slave I2C Controller 5 Interrupt Event		168
	6	I2C Controller6	I2C	No	Slave I2C Controller 6 Interrupt Event		169
	7	I2C Controller7	I2C	No	Slave I2C Controller 7 Interrupt Event		170
	8-31	Reserved					
GIRQ14	0	DMA Controller	DMA0	No	DMA Controller - Channel 0 Interrupt Event	6	24

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	1	DMA Controller	DMA1	No	DMA Controller - Channel 1 Interrupt Event		25
	2	DMA Controller	DMA2	No	DMA Controller - Channel 2 Interrupt Event		26
	3	DMA Controller	DMA3	No	DMA Controller - Channel 3 Interrupt Event		27
	4	DMA Controller	DMA4	No	DMA Controller - Channel 4 Interrupt Event		28
	5	DMA Controller	DMA5	No	DMA Controller - Channel 5 Interrupt Event		29
	6	DMA Controller	DMA6	No	DMA Controller - Channel 6 Interrupt Event		30
	7	DMA Controller	DMA7	No	DMA Controller - Channel 7 Interrupt Event		31
	8	DMA Controller	DMA8	No	DMA Controller - Channel 8 Interrupt Event		32
	9	DMA Controller	DMA9	No	DMA Controller - Channel 9 Interrupt Event		33
	10	DMA Controller	DMA10	No	DMA Controller - Channel 10 Interrupt Event		34
	11	DMA Controller	DMA11	No	DMA Controller - Channel 11 Interrupt Event		35
	12-31	Reserved					
GIRQ15	0	UART 0	UART	No	UART Interrupt Event	7	40
	1	UART 1	UART	No	UART Interrupt Event		41
	2-4	Reserved					42
	4	UART2	UART	No	UART Interrupt Event		44
	5-31	Reserved					45
GIRQ17	0	Reserved					70
	1	TACH 0	TACH	No	Tachometer 0 Interrupt Event		71
	2	TACH 1	TACH	No	Tachometer 1 Interrupt Event		72
	3-7	Reserved					
	8	ADC Controller	ADC_Single_Int	No	ADC Controller - Single-Sample ADC Conversion Event		78
	9	ADC Controller	ADC_Repeat_Int	No	ADC Controller - Repeat-Sample ADC Conversion Event		79
	10-12	Reserved					
	13	Breathing LED 0	PWM_WDT	No	Blinking LED 0 Watchdog Event		83
	14	Breathing LED 1	PWM_WDT	No	Blinking LED 1 Watchdog Event		84
	15-31	Reserved					
GIRQ18	0	Reserved				10	

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	1	Quad Master SPI Controller	QMSPI_INT	No	Master SPI Controller Requires Servicing		91
	2-19	Reserved					
	20	Capture Compare Timer	CAPTURE TIMER	No	CCT Counter Event		146
	21	Capture Compare Timer	CAPTURE 0	No	CCT Capture 0 Event		147
	22	Capture Compare Timer	CAPTURE 1	No	CCT Capture 1 Event		148
	23	Capture Compare Timer	CAPTURE 2	No	CCT Capture 2 Event		149
	24	Capture Compare Timer	CAPTURE 3	No	CCT Capture 3 Event		150
	25	Capture Compare Timer	CAPTURE 4	No	CCT Capture 4 Event		151
	26	Capture Compare Timer	CAPTURE 5	No	CCT Capture 5 Event		152
	27	Capture Compare Timer	COMPARE 0	No	CCT Compare 0 Event		153
	28	Capture Compare Timer	COMPARE 1	No	CCT Compare 1 Event		154
	29-31	Reserved					
GIRQ19		Reserved				11	103
GIRQ20	0-2	Reserved					
	3	OTP	READY_INTR	No	OTP ready interrupt		173
	4-31	Reserved					
GIRQ21	0-1	Reserved				13	
	2	WDT	WDT_INT	Yes	Watch Dog Timer Interupt		171
	3	Week Alarm	WEEK_ALARM_INT	Yes	Week Alarm Interrupt.		114
	4	Week Alarm	SUB-_WEEK_ALARM_INT	Yes	Sub-Week Alarm Interrupt		115
	5	Week Alarm	ONE_SECOND	Yes	Week Alarm - One Second Interrupt		116
	6	Week Alarm	SUB_SECOND	Yes	Week Alarm - Sub-second Interrupt		117
	7	Week Alarm	SYSPWR_PRES	Yes	System power present pin interrupt		118
	8	RTC	RTC	Yes	Real Time Clock Interrupt		119
	9	RTC	RTC ALARM	Yes	Real Time Clock Alarm Interrupt		120
	10	Reserved					
	11	VBAT-Powered Control Interface	VCI_IN0	Yes	VCI_IN0 Active-low Input Pin Interrupt		122

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	14	VBAT-Powered Control Interface	VCI_IN3	Yes	VCI_IN3 Active-low Input Pin Interrupt		125
	15-31	Reserved					
GIRQ22	0	Reserved				N/A	N/A
	1	SMB-I2C Controller0	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.0 START Detected		
	2	SMB-I2C Controller1	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.1 START Detected		
	3	SMB-I2C Controller2	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.2 START Detected		
	4	SMB-I2C Controller3	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.3 START Detected		
	5	SMB-I2C Controller4	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.4 START Detected		
	6	I2C Controller5	I2C	Yes	Slave I2C Controller 5 Wake Event		
	7	I2C Controller6	I2C	Yes	Slave I2C Controller 6 Wake Event		
	8	I2C Controller7	I2C	Yes	Slave I2C Controller 7 Wake Event		
	6-31	Reserved					
GIRQ23	0	16-Bit Basic Timer 0	Timer_16_0	No	Basic Timer Event	14	136
	1	16-Bit Basic Timer 1	Timer_16_1	No	Basic Timer Event		137
	2-3	Reserved					
	4	32-Bit Basic Timer 0	Timer_32_0	No	Basic Timer Event		140
	5	32-Bit Basic Timer 1	Timer_32_1	No	Basic Timer Event		141
	6-9	Reserved					
	10	RTOS Timer	RTOS_TIMER	Yes	32-bit RTOS Timer Event		111
	11	RTOS Timer	SWI_0	No	Soft Interrupt request 0		
	12	RTOS Timer	SWI_1	No	Soft Interrupt request 1		
	13	RTOS Timer	SWI_2	No	Soft Interrupt request 2		
	14	RTOS Timer	SWI_3	No	Soft Interrupt request 3		
	15	Reserved					
	16	Hibernation Timer0	HTIMER	Yes	Hibernation Timer Event		112
	17	Hibernation Timer1	HTIMER	Yes	Hibernation Timer Event		113

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	18-31	Reserved					
GIRQ24		Reserved					N/A
GIRQ25		Reserved					N/A
GIRQ26	0-7	Reserved				17	N/A
	8	GPIO250	GPIO Event	Yes	GPIO Interrupt Event		
	9-10	Reserved					
	11	GPIO253	GPIO Event	Yes	GPIO Interrupt Event		
	12-31	Reserved					

Note: Registers and bits associated with GPIOs not implemented are Reserved. Please refer to Section 2.3, "Pin List" for GPIOs implemented in the chip.

3.5 GPIO Register Assignments

All GPIOs except the below come up in default GPIO Input/output/interrupt disabled state. Pin control register defaults to 0x00008040.

TABLE 3-4: GPIO PIN CONTROL DEFAULT VALUES

GPIO	Pin control register value	Default function
GPIO000	0x00001040	VCI_IN
GPIO163	0x00001040	VCI_IN
GPIO170	0x00000041	JTAG_STRAP BS (input, pull up)
GPIO250	0x00001240	VCI_OUT

3.6 Register Map

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Watchdog Timer	0	WDT Load Register		40000400
Watchdog Timer	0	WDT Control Register		40000404
Watchdog Timer	0	WDT Kick Register		40000408
Watchdog Timer	0	WDT Count Register		4000040C
Watchdog Timer	0	WDT Status Register		40000410
Watchdog Timer	0	WDT Interrupt Enable Register		40000414
16-bit Basic Timer	0	Timer Count Register		40000C00
16-bit Basic Timer	0	Timer Preload Register		40000C04
16-bit Basic Timer	0	Timer Status Register		40000C08
16-bit Basic Timer	0	Timer Int Enable Register		40000C0C
16-bit Basic Timer	0	Timer Control Register		40000C10
16-bit Basic Timer	1	Timer Count Register		40000C20
16-bit Basic Timer	1	Timer Preload Register		40000C24
16-bit Basic Timer	1	Timer Status Register		40000C28
16-bit Basic Timer	1	Timer Int Enable Register		40000C2C
16-bit Basic Timer	1	Timer Control Register		40000C30
32-bit Basic Timer	0	Timer Count Register		40000C80
32-bit Basic Timer	0	Timer Preload Register		40000C84
32-bit Basic Timer	0	Timer Status Register		40000C88
32-bit Basic Timer	0	Timer Int Enable Register		40000C8C
32-bit Basic Timer	0	Timer Control Register		40000C90
32-bit Basic Timer	1	Timer Count Register		40000CA0
32-bit Basic Timer	1	Timer Preload Register		40000CA4
32-bit Basic Timer	1	Timer Status Register		40000CA8
32-bit Basic Timer	1	Timer Int Enable Register		40000CAC
32-bit Basic Timer	1	Timer Control Register		40000CB0
Capture Compare Timer	0	Capture and Compare Timer Control Register		40001000
Capture Compare Timer	0	Capture Control 0 Register		40001004
Capture Compare Timer	0	Capture Control 1 Register		40001008

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Capture Compare Timer	0	Free Running Timer Register		4000100C
Capture Compare Timer	0	Capture 0 Register		40001010
Capture Compare Timer	0	Capture 1 Register		40001014
Capture Compare Timer	0	Capture 2 Register		40001018
Capture Compare Timer	0	Capture 3 Register		4000101C
Capture Compare Timer	0	Capture 4 Register		40001020
Capture Compare Timer	0	Capture 5 Register		40001024
Capture Compare Timer	0	Compare 0 Register		40001028
Capture Compare Timer	0	Compare 1 Register		4000102C
Capture Compare Timer	0	ICT Mux Select Register		40001030
DMA Controller	0	DMA Main Control Register		40002400
DMA Controller	0	DMA Data Packet Register		40002404
DMA Controller	0	TEST		40002408
DMA Channel	0	DMA Channel N Activate Register		40002440
DMA Channel	0	DMA Channel N Memory Start Address Register		40002444
DMA Channel	0	DMA Channel N Memory End Address Register		40002448
DMA Channel	0	DMA Channel N Device Address		4000244C
DMA Channel	0	DMA Channel N Control Register		40002450
DMA Channel	0	DMA Channel N Interrupt Status Register		40002454
DMA Channel	0	DMA Channel N Interrupt Enable Register		40002458
DMA Channel	0	TEST		4000245C
DMA Channel	0	Channel N CRC Enable Register		40002460
DMA Channel	0	Channel N CRC Data Register		40002464
DMA Channel	0	Channel N CRC Post Status Register		40002468
DMA Channel	0	TEST		4000246C
DMA Channel	1	DMA Channel N Activate Register		40002480
DMA Channel	1	DMA Channel N Memory Start Address Register		40002484
DMA Channel	1	DMA Channel N Memory End Address Register		40002488
DMA Channel	1	DMA Channel N Device Address		4000248C
DMA Channel	1	DMA Channel N Control Register		40002490
DMA Channel	1	DMA Channel N Interrupt Status Register		40002494
DMA Channel	1	DMA Channel N Interrupt Enable Register		40002498
DMA Channel	1	TEST		4000249C
DMA Channel	1	Channel N Fill Enable Register		400024A0
DMA Channel	1	Channel N Fill Data Register		400024A4
DMA Channel	1	Channel N Fill Status Register		400024A8
DMA Channel	1	TEST		400024AC
DMA Channel	2	DMA Channel N Activate Register		400024C0
DMA Channel	2	DMA Channel N Memory Start Address Register		400024C4
DMA Channel	2	DMA Channel N Memory End Address Register		400024C8
DMA Channel	2	DMA Channel N Device Address		400024CC
DMA Channel	2	DMA Channel N Control Register		400024D0
DMA Channel	2	DMA Channel N Interrupt Status Register		400024D4
DMA Channel	2	DMA Channel N Interrupt Enable Register		400024D8

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
DMA Channel	2	TEST		400024DC
DMA Channel	3	DMA Channel N Activate Register		40002500
DMA Channel	3	DMA Channel N Memory Start Address Register		40002504
DMA Channel	3	DMA Channel N Memory End Address Register		40002508
DMA Channel	3	DMA Channel N Device Address		4000250C
DMA Channel	3	DMA Channel N Control Register		40002510
DMA Channel	3	DMA Channel N Interrupt Status Register		40002514
DMA Channel	3	DMA Channel N Interrupt Enable Register		40002518
DMA Channel	3	TEST		4000251C
DMA Channel	4	DMA Channel N Activate Register		40002540
DMA Channel	4	DMA Channel N Memory Start Address Register		40002544
DMA Channel	4	DMA Channel N Memory End Address Register		40002548
DMA Channel	4	DMA Channel N Device Address		4000254C
DMA Channel	4	DMA Channel N Control Register		40002550
DMA Channel	4	DMA Channel N Interrupt Status Register		40002554
DMA Channel	4	DMA Channel N Interrupt Enable Register		40002558
DMA Channel	4	TEST		4000255C
DMA Channel	5	DMA Channel N Activate Register		40002580
DMA Channel	5	DMA Channel N Memory Start Address Register		40002584
DMA Channel	5	DMA Channel N Memory End Address Register		40002588
DMA Channel	5	DMA Channel N Device Address		4000258C
DMA Channel	5	DMA Channel N Control Register		40002590
DMA Channel	5	DMA Channel N Interrupt Status Register		40002594
DMA Channel	5	DMA Channel N Interrupt Enable Register		40002598
DMA Channel	5	TEST		4000259C
DMA Channel	6	DMA Channel N Activate Register		400025C0
DMA Channel	6	DMA Channel N Memory Start Address Register		400025C4
DMA Channel	6	DMA Channel N Memory End Address Register		400025C8
DMA Channel	6	DMA Channel N Device Address		400025CC
DMA Channel	6	DMA Channel N Control Register		400025D0
DMA Channel	6	DMA Channel N Interrupt Status Register		400025D4
DMA Channel	6	DMA Channel N Interrupt Enable Register		400025D8
DMA Channel	6	TEST		400025DC
DMA Channel	7	DMA Channel N Activate Register		40002600
DMA Channel	7	DMA Channel N Memory Start Address Register		40002604
DMA Channel	7	DMA Channel N Memory End Address Register		40002608
DMA Channel	7	DMA Channel N Device Address		4000260C
DMA Channel	7	DMA Channel N Control Register		40002610
DMA Channel	7	DMA Channel N Interrupt Status Register		40002614
DMA Channel	7	DMA Channel N Interrupt Enable Register		40002618
DMA Channel	7	TEST		4000261C
DMA Channel	8	DMA Channel N Activate Register		40002640
DMA Channel	8	DMA Channel N Memory Start Address Register		40002644
DMA Channel	8	DMA Channel N Memory End Address Register		40002648

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
DMA Channel	8	DMA Channel N Device Address		4000264C
DMA Channel	8	DMA Channel N Control Register		40002650
DMA Channel	8	DMA Channel N Interrupt Status Register		40002654
DMA Channel	8	DMA Channel N Interrupt Enable Register		40002658
DMA Channel	8	TEST		4000265C
DMA Channel	9	DMA Channel N Activate Register		40002680
DMA Channel	9	DMA Channel N Memory Start Address Register		40002684
DMA Channel	9	DMA Channel N Memory End Address Register		40002688
DMA Channel	9	DMA Channel N Device Address		4000268C
DMA Channel	9	DMA Channel N Control Register		40002690
DMA Channel	9	DMA Channel N Interrupt Status Register		40002694
DMA Channel	9	DMA Channel N Interrupt Enable Register		40002698
DMA Channel	9	TEST		4000269C
DMA Channel	10	DMA Channel N Activate Register		400026C0
DMA Channel	10	DMA Channel N Memory Start Address Register		400026C4
DMA Channel	10	DMA Channel N Memory End Address Register		400026C8
DMA Channel	10	DMA Channel N Device Address		400026CC
DMA Channel	10	DMA Channel N Control Register		400026D0
DMA Channel	10	DMA Channel N Interrupt Status Register		400026D4
DMA Channel	10	DMA Channel N Interrupt Enable Register		400026D8
DMA Channel	10	TEST		400026DC
DMA Channel	11	DMA Channel N Activate Register		40002700
DMA Channel	11	DMA Channel N Memory Start Address Register		40002704
DMA Channel	11	DMA Channel N Memory End Address Register		40002708
DMA Channel	11	DMA Channel N Device Address		4000270C
DMA Channel	11	DMA Channel N Control Register		40002710
DMA Channel	11	DMA Channel N Interrupt Status Register		40002714
DMA Channel	11	DMA Channel N Interrupt Enable Register		40002718
DMA Channel	11	TEST		4000271C
I2C-SMB	0	Control Register		40004000
I2C-SMB	0	Status Register		40004000
I2C-SMB	0	Own Address Register		40004004
I2C-SMB	0	Data Register		40004008
I2C-SMB	0	Master Command Register		4000400C
I2C-SMB	0	Slave Command Register		40004010
I2C-SMB	0	PEC Register		40004014
I2C-SMB	0	Repeated START Hold Time Register		40004018
I2C-SMB	0	Completion Register		40004020
I2C-SMB	0	Idle Scaling Register		40004024
I2C-SMB	0	Configuration Register		40004028
I2C-SMB	0	Bus Clock Register		4000402C
I2C-SMB	0	Block ID Register		40004030
I2C-SMB	0	Revision Register		40004034
I2C-SMB	0	Bit-Bang Control Register		40004038

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
I2C-SMB	0	TEST		4000403C
I2C-SMB	0	Data Timing Register		40004040
I2C-SMB	0	Time-Out Scaling Register		40004044
I2C-SMB	0	Slave Transmit Buffer Register		40004048
I2C-SMB	0	Slave Receive Buffer Register		4000404C
I2C-SMB	0	Master Transmit Buffer Register		40004050
I2C-SMB	0	Master Receive Buffer Register		40004054
I2C-SMB	0	TEST		40004058
I2C-SMB	0	TEST		4000405C
I2C-SMB	0	Wake Status Register		40004060
I2C-SMB	0	Wake Enable Register		40004064
I2C-SMB	0	TEST		40004068
I2C-SMB	0	Slave address		4000406C
I2C-SMB	0	Promiscuous Interrupt		40004070
I2C-SMB	0	Promiscuous Interrupt Enable		40004074
I2C-SMB	0	Promiscuous Control		40004078
I2C-SMB	1	Control Register		40004400
I2C-SMB	1	Status Register		40004400
I2C-SMB	1	Own Address Register		40004404
I2C-SMB	1	Data Register		40004408
I2C-SMB	1	Master Command Register		4000440C
I2C-SMB	1	Slave Command Register		40004410
I2C-SMB	1	PEC Register		40004414
I2C-SMB	1	Repeated START Hold Time Register		40004418
I2C-SMB	1	Completion Register		40004420
I2C-SMB	1	Idle Scaling Register		40004424
I2C-SMB	1	Configuration Register		40004428
I2C-SMB	1	Bus Clock Register		4000442C
I2C-SMB	1	Block ID Register		40004430
I2C-SMB	1	Revision Register		40004434
I2C-SMB	1	Bit-Bang Control Register		40004438
I2C-SMB	1	TEST		4000443C
I2C-SMB	1	Data Timing Register		40004440
I2C-SMB	1	Time-Out Scaling Register		40004444
I2C-SMB	1	Slave Transmit Buffer Register		40004448
I2C-SMB	1	Slave Receive Buffer Register		4000444C
I2C-SMB	1	Master Transmit Buffer Register		40004450
I2C-SMB	1	Master Receive Buffer Register		40004454
I2C-SMB	1	TEST		40004458
I2C-SMB	1	TEST		4000445C
I2C-SMB	1	Wake Status Register		40004460
I2C-SMB	1	Wake Enable Register		40004464
I2C-SMB	1	TEST		40004468
I2C-SMB	1	Slave address		4000446C

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
I2C-SMB	1	Promiscuous Interrupt		40004470
I2C-SMB	1	Promiscuous Interrupt Enable		40004474
I2C-SMB	1	Promiscuous Control		40004478
I2C-SMB	2	Control Register		40004800
I2C-SMB	2	Status Register		40004800
I2C-SMB	2	Own Address Register		40004804
I2C-SMB	2	Data Register		40004808
I2C-SMB	2	Master Command Register		4000480C
I2C-SMB	2	Slave Command Register		40004810
I2C-SMB	2	PEC Register		40004814
I2C-SMB	2	Repeated START Hold Time Register		40004818
I2C-SMB	2	Completion Register		40004820
I2C-SMB	2	Idle Scaling Register		40004824
I2C-SMB	2	Configuration Register		40004828
I2C-SMB	2	Bus Clock Register		4000482C
I2C-SMB	2	Block ID Register		40004830
I2C-SMB	2	Revision Register		40004834
I2C-SMB	2	Bit-Bang Control Register		40004838
I2C-SMB	2	TEST		4000483C
I2C-SMB	2	Data Timing Register		40004840
I2C-SMB	2	Time-Out Scaling Register		40004844
I2C-SMB	2	Slave Transmit Buffer Register		40004848
I2C-SMB	2	Slave Receive Buffer Register		4000484C
I2C-SMB	2	Master Transmit Buffer Register		40004850
I2C-SMB	2	Master Receive Buffer Register		40004854
I2C-SMB	2	TEST		40004858
I2C-SMB	2	TEST		4000485C
I2C-SMB	2	Wake Status Register		40004860
I2C-SMB	2	Wake Enable Register		40004864
I2C-SMB	2	TEST		40004868
I2C-SMB	2	Slave address		4000486C
I2C-SMB	2	Promiscuous Interrupt		40004870
I2C-SMB	2	Promiscuous Interrupt Enable		40004874
I2C-SMB	2	Promiscuous Control		40004878
I2C-SMB	3	Control Register		40004C00
I2C-SMB	3	Status Register		40004C00
I2C-SMB	3	Own Address Register		40004C04
I2C-SMB	3	Data Register		40004C08
I2C-SMB	3	Master Command Register		40004C0C
I2C-SMB	3	Slave Command Register		40004C10
I2C-SMB	3	PEC Register		40004C14
I2C-SMB	3	Repeated START Hold Time Register		40004C18
I2C-SMB	3	Completion Register		40004C20
I2C-SMB	3	Idle Scaling Register		40004C24

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
I2C-SMB	3	Configuration Register		40004C28
I2C-SMB	3	Bus Clock Register		40004C2C
I2C-SMB	3	Block ID Register		40004C30
I2C-SMB	3	Revision Register		40004C34
I2C-SMB	3	Bit-Bang Control Register		40004C38
I2C-SMB	3	TEST		40004C3C
I2C-SMB	3	Data Timing Register		40004C40
I2C-SMB	3	Time-Out Scaling Register		40004C44
I2C-SMB	3	Slave Transmit Buffer Register		40004C48
I2C-SMB	3	Slave Receive Buffer Register		40004C4C
I2C-SMB	3	Master Transmit Buffer Register		40004C50
I2C-SMB	3	Master Receive Buffer Register		40004C54
I2C-SMB	3	TEST		40004C58
I2C-SMB	3	TEST		40004C5C
I2C-SMB	3	Wake Status Register		40004C60
I2C-SMB	3	Wake Enable Register		40004C64
I2C-SMB	3	TEST		40004C68
I2C-SMB	3	Slave address		40004C6C
I2C-SMB	3	Promiscuous Interrupt		40004C70
I2C-SMB	3	Promiscuous Interrupt Enable		40004C74
I2C-SMB	3	Promiscuous Control		40004C78
I2C-SMB	4	Control Register		40005000
I2C-SMB	4	Status Register		40005000
I2C-SMB	4	Own Address Register		40005004
I2C-SMB	4	Data Register		40005008
I2C-SMB	4	Master Command Register		4000500C
I2C-SMB	4	Slave Command Register		40005010
I2C-SMB	4	PEC Register		40005014
I2C-SMB	4	Repeated START Hold Time Register		40005018
I2C-SMB	4	Completion Register		40005020
I2C-SMB	4	Idle Scaling Register		40005024
I2C-SMB	4	Configuration Register		40005028
I2C-SMB	4	Bus Clock Register		4000502C
I2C-SMB	4	Block ID Register		40005030
I2C-SMB	4	Revision Register		40005034
I2C-SMB	4	Bit-Bang Control Register		40005038
I2C-SMB	4	TEST		4000503C
I2C-SMB	4	Data Timing Register		40005040
I2C-SMB	4	Time-Out Scaling Register		40005044
I2C-SMB	4	Slave Transmit Buffer Register		40005048
I2C-SMB	4	Slave Receive Buffer Register		4000504C
I2C-SMB	4	Master Transmit Buffer Register		40005050
I2C-SMB	4	Master Receive Buffer Register		40005054
I2C-SMB	4	TEST		40005058

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
I2C-SMB	4	TEST		4000505C
I2C-SMB	4	Wake Status Register		40005060
I2C-SMB	4	Wake Enable Register		40005064
I2C-SMB	4	TEST		40005068
I2C-SMB	4	Slave address		4000506C
I2C-SMB	4	Promiscuous Interrupt		40005070
I2C-SMB	4	Promiscuous Interrupt Enable		40005074
I2C-SMB	4	Promiscuous Control		40005078
I2C	0	Control Register		40005100
I2C	0	Status Register		40005100
I2C	0	Own Address Register		40005104
I2C	0	Data Register		40005108
I2C	0	Repeated START Hold Time Register		40005118
I2C	0	Completion Register		40005120
I2C	0	Configuration Register		40005128
I2C	0	Bus Clock Register		4000512C
I2C	0	Block ID Register		40005130
I2C	0	Revision Register		40005134
I2C	0	Bit-Bang Control Register		40005138
I2C	0	TEST		4000513C
I2C	0	Data Timing Register		40005140
I2C	0	Time-Out Scaling Register		40005144
I2C	0	TEST		40005158
I2C	0	TEST		4000515C
I2C	0	Wake Status Register		40005160
I2C	0	Wake Enable Register		40005164
I2C	0	TEST		40005168
I2C	0	Slave address		4000516C
I2C	0	Promiscuous Interrupt		40005170
I2C	0	Promiscuous Interrupt Enable		40005174
I2C	0	Promiscuous Control		40005178
I2C	1	Control Register		40005200
I2C	1	Status Register		40005200
I2C	1	Own Address Register		40005204
I2C	1	Data Register		40005208
I2C	1	Repeated START Hold Time Register		40005218
I2C	1	Completion Register		40005220
I2C	1	Configuration Register		40005228
I2C	1	Bus Clock Register		4000522C
I2C	1	Block ID Register		40005230
I2C	1	Revision Register		40005234
I2C	1	Bit-Bang Control Register		40005238
I2C	1	TEST		4000523C
I2C	1	Data Timing Register		40005240

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
I2C	1	Time-Out Scaling Register		40005244
I2C	1	TEST		40005258
I2C	1	TEST		4000525C
I2C	1	Wake Status Register		40005260
I2C	1	Wake Enable Register		40005264
I2C	1	TEST		40005268
I2C	1	Slave address		4000526C
I2C	1	Promiscuous Interrupt		40005270
I2C	1	Promiscuous Interrupt Enable		40005274
I2C	1	Promiscuous Control		40005278
I2C	2	Control Register		40005300
I2C	2	Status Register		40005300
I2C	2	Own Address Register		40005304
I2C	2	Data Register		40005308
I2C	2	Repeated START Hold Time Register		40005318
I2C	2	Completion Register		40005320
I2C	2	Configuration Register		40005328
I2C	2	Bus Clock Register		4000532C
I2C	2	Block ID Register		40005330
I2C	2	Revision Register		40005334
I2C	2	Bit-Bang Control Register		40005338
I2C	2	TEST		4000533C
I2C	2	Data Timing Register		40005340
I2C	2	Time-Out Scaling Register		40005344
I2C	2	TEST		40005358
I2C	2	TEST		4000535C
I2C	2	Wake Status Register		40005360
I2C	2	Wake Enable Register		40005364
I2C	2	TEST		40005368
I2C	2	Slave address		4000536C
I2C	2	Promiscuous Interrupt		40005370
I2C	2	Promiscuous Interrupt Enable		40005374
I2C	2	Promiscuous Control		40005378
QMSPI	0	QMSPI Mode Register		40070000
QMSPI	0	QMSPI Control Register		40070004
QMSPI	0	QMSPI Execute Register		40070008
QMSPI	0	QMSPI Interface Control Register		4007000C
QMSPI	0	QMSPI Status Register		40070010
QMSPI	0	QMSPI Buffer Count Status Register		40070014
QMSPI	0	QMSPI Interrupt Enable Register		40070018
QMSPI	0	QMSPI Buffer Count Trigger Register		4007001C
QMSPI	0	QMSPI Transmit Buffer Register		40070020
QMSPI	0	QMSPI Receive Buffer Register		40070024
QMSPI	0	QMSPI Chip Select Timing Register		40070028

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
QMSPI	0	QMSPI Description Buffer 0 Register		40070030
QMSPI	0	QMSPI Description Buffer 1 Register		40070034
QMSPI	0	QMSPI Description Buffer 2 Register		40070038
QMSPI	0	QMSPI Description Buffer 3 Register		4007003C
QMSPI	0	QMSPI Description Buffer 4 Register		40070040
QMSPI	0	QMSPI Description Buffer 5 Register		40070044
QMSPI	0	QMSPI Description Buffer 6 Register		40070048
QMSPI	0	QMSPI Description Buffer 7 Register		4007004C
QMSPI	0	QMSPI Description Buffer 8 Register		40070050
QMSPI	0	QMSPI Description Buffer 9 Register		40070054
QMSPI	0	QMSPI Description Buffer 10 Register		40070058
QMSPI	0	QMSPI Description Buffer 11 Register		4007005C
QMSPI	0	QMSPI Description Buffer 12 Register		40070060
QMSPI	0	QMSPI Description Buffer 13 Register		40070064
QMSPI	0	QMSPI Description Buffer 14 Register		40070068
QMSPI	0	QMSPI Description Buffer 15 Register		4007006C
16-bit PWM	0	PWMx Counter ON Time Register		40005800
16-bit PWM	0	PWMx Counter OFF Time Register		40005804
16-bit PWM	0	PWMx Configuration Register		40005808
16-bit PWM	0	TEST		4000580C
16-bit PWM	2	PWMx Counter ON Time Register		40005820
16-bit PWM	2	PWMx Counter OFF Time Register		40005824
16-bit PWM	2	PWMx Configuration Register		40005828
16-bit PWM	2	TEST		4000582C
16-bit PWM	3	PWMx Counter ON Time Register		40005830
16-bit PWM	3	PWMx Counter OFF Time Register		40005834
16-bit PWM	3	PWMx Configuration Register		40005838
16-bit PWM	3	TEST		4000583C
16-bit PWM	5	PWMx Counter ON Time Register		40005850
16-bit PWM	5	PWMx Counter OFF Time Register		40005854
16-bit PWM	5	PWMx Configuration Register		40005858
16-bit PWM	5	TEST		4000585C
16-bit PWM	6	PWMx Counter ON Time Register		40005860
16-bit PWM	6	PWMx Counter OFF Time Register		40005864
16-bit PWM	6	PWMx Configuration Register		40005868
16-bit PWM	6	TEST		4000586C
16-bit PWM	7	PWMx Counter ON Time Register		40005870
16-bit PWM	7	PWMx Counter OFF Time Register		40005874
16-bit PWM	7	PWMx Configuration Register		40005878
16-bit PWM	7	TEST		4000587C
16-bit Tach	0	TACHx Control Register		40006000
16-bit Tach	0	TACHx Status Register		40006004
16-bit Tach	0	TACHx High Limit Register		40006008
16-bit Tach	0	TACHx Low Limit Register		4000600C

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
16-bit Tach	1	TACHx Control Register		40006010
16-bit Tach	1	TACHx Status Register		40006014
16-bit Tach	1	TACHx High Limit Register		40006018
16-bit Tach	1	TACHx Low Limit Register		4000601C
RTOS Timer	0	RTOS Timer Count Register		40007400
RTOS Timer	0	RTOS Timer Preload Register		40007404
RTOS Timer	0	RTOS Timer Control Register		40007408
RTOS Timer	0	Soft Interrupt Register		4000740C
ADC	0	ADC Control Register		40007C00
ADC	0	ADC Delay Register		40007C04
ADC	0	ADC Status Register		40007C08
ADC	0	ADC Single Register		40007C0C
ADC	0	ADC Repeat Register		40007C10
ADC	0	ADC Channel 0 Reading Register		40007C14
ADC	0	ADC Channel 1 Reading Register		40007C18
ADC	0	ADC Channel 2 Reading Register		40007C1C
ADC	0	ADC Channel 3 Reading Register		40007C20
ADC	0	ADC Channel 4 Reading Register		40007C24
ADC	0	ADC Channel 5 Reading Register		40007C28
ADC	0	ADC Channel 6 Reading Register		40007C2C
ADC	0	ADC Channel 7 Reading Register		40007C30
ADC	0	ADC Channel 8 Reading Register		40007C34
ADC	0	ADC Channel 9 Reading Register		40007C38
ADC	0	ADC Channel 10 Reading Register		40007C3C
ADC	0	ADC Channel 11 Reading Register		40007C40
ADC	0	ADC Configuration Register		40007C7C
ADC	0	VREF Channel Register		40007C80
ADC	0	VREF Control Register		40007C84
ADC	0	SAR ADC Control Register		40007C88
ADC	0	SAR ADC Config Register		40007C8C
TFDP	0	Debug Data Register		40008C00
TFDP	0	Debug Control Register		40008C04
Hibernation Timer	0	HTimer Preload Register		40009800
Hibernation Timer	0	HTimer Control Register		40009804
Hibernation Timer	0	HTimer Count Register		40009808
Hibernation Timer	1	HTimer Preload Register		40009820
Hibernation Timer	1	HTimer Control Register		40009824
Hibernation Timer	1	HTimer Count Register		40009828
VBAT Register Bank	0	Power-Fail and Reset Status Register		4000A400
VBAT Register Bank	0	TEST		4000A404
VBAT Register Bank	0	Clock Enable Register		4000A408
VBAT Register Bank	0	TEST		4000A40C
VBAT Register Bank	0	TEST		4000A410
VBAT Register Bank	0	TEST		4000A414

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
VBAT Register Bank	0	TEST		4000A41C
VBAT Register Bank	0	Monotonic Counter Register		4000A420
VBAT Register Bank	0	Counter HiWord Register		4000A424
VBAT Register Bank	0	TEST		4000A428
VBAT Register Bank	0	TEST		4000A42C
VBAT Powered RAM	0	Registers		4000A800
Week Timer	0	Control Register		4000AC80
Week Timer	0	Week Alarm Counter Register		4000AC84
Week Timer	0	Week Timer Compare Register		4000AC88
Week Timer	0	Clock Divider Register		4000AC8C
Week Timer	0	Sub-Second Programmable Interrupt Select Register		4000AC90
Week Timer	0	Sub-Week Control Register		4000AC94
Week Timer	0	Sub-Week Alarm Counter Register		4000AC98
Week Timer	0	BGPO Data Register		4000AC9C
Week Timer	0	BGPO Power Register		4000ACA0
Week Timer	0	BGPO Reset Register		4000ACA4
VBAT-Powered Control Interface	0	VCI Register		4000AE00
VBAT-Powered Control Interface	0	Latch Enable Register		4000AE04
VBAT-Powered Control Interface	0	Latch Resets Register		4000AE08
VBAT-Powered Control Interface	0	VCI Input Enable Register		4000AE0C
VBAT-Powered Control Interface	0	Holdoff Count Register		4000AE10
VBAT-Powered Control Interface	0	VCI Polarity Register		4000AE14
VBAT-Powered Control Interface	0	VCI Posedge Detect Register		4000AE18
VBAT-Powered Control Interface	0	VCI Negedge Detect Register		4000AE1C
VBAT-Powered Control Interface	0	VCI Buffer Enable Register		4000AE20
Blinking-Breathing PWM	0	LED Configuration Register		4000B800
Blinking-Breathing PWM	0	LED Limits Register		4000B804
Blinking-Breathing PWM	0	LED Delay Register		4000B808
Blinking-Breathing PWM	0	LED Update Stepsize Register		4000B80C
Blinking-Breathing PWM	0	LED Update Interval Register		4000B810
Blinking-Breathing PWM	0	LED Output Delay		4000B814
Blinking-Breathing PWM	1	LED Configuration Register		4000B900
Blinking-Breathing PWM	1	LED Limits Register		4000B904
Blinking-Breathing PWM	1	LED Delay Register		4000B908
Blinking-Breathing PWM	1	LED Update Stepsize Register		4000B90C
Blinking-Breathing PWM	1	LED Update Interval Register		4000B910
Blinking-Breathing PWM	1	LED Output Delay		4000B914

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Interrupt Aggregator	0	GIRQ8 Source Register		4000E000
Interrupt Aggregator	0	GIRQ8 Enable Set Register		4000E004
Interrupt Aggregator	0	GIRQ8 Result Register		4000E008
Interrupt Aggregator	0	GIRQ8 Enable Clear Register		4000E00C
Interrupt Aggregator	0	GIRQ9 Source Register		4000E014
Interrupt Aggregator	0	GIRQ9 Enable Set Register		4000E018
Interrupt Aggregator	0	GIRQ9 Result Register		4000E01C
Interrupt Aggregator	0	GIRQ9 Enable Clear Register		4000E020
Interrupt Aggregator	0	GIRQ10 Source Register		4000E028
Interrupt Aggregator	0	GIRQ10 Enable Set Register		4000E02C
Interrupt Aggregator	0	GIRQ10 Result Register		4000E030
Interrupt Aggregator	0	GIRQ10 Enable Clear Register		4000E034
Interrupt Aggregator	0	GIRQ11 Source Register		4000E03C
Interrupt Aggregator	0	GIRQ11 Enable Set Register		4000E040
Interrupt Aggregator	0	GIRQ11 Result Register		4000E044
Interrupt Aggregator	0	GIRQ11 Enable Clear Register		4000E048
Interrupt Aggregator	0	GIRQ12 Source Register		4000E050
Interrupt Aggregator	0	GIRQ12 Enable Set Register		4000E054
Interrupt Aggregator	0	GIRQ12 Result Register		4000E058
Interrupt Aggregator	0	GIRQ12 Enable Clear Register		4000E05C
Interrupt Aggregator	0	GIRQ13 Source Register		4000E064
Interrupt Aggregator	0	GIRQ13 Enable Set Register		4000E068
Interrupt Aggregator	0	GIRQ13 Result Register		4000E06C
Interrupt Aggregator	0	GIRQ13 Enable Clear Register		4000E070
Interrupt Aggregator	0	GIRQ14 Source Register		4000E078
Interrupt Aggregator	0	GIRQ14 Enable Set Register		4000E07C
Interrupt Aggregator	0	GIRQ14 Result Register		4000E080
Interrupt Aggregator	0	GIRQ14 Enable Clear Register		4000E084
Interrupt Aggregator	0	GIRQ15 Source Register		4000E08C
Interrupt Aggregator	0	GIRQ15 Enable Set Register		4000E090
Interrupt Aggregator	0	GIRQ15 Result Register		4000E094
Interrupt Aggregator	0	GIRQ15 Enable Clear Register		4000E098
Interrupt Aggregator	0	GIRQ16 Source Register		4000E0A0
Interrupt Aggregator	0	GIRQ16 Enable Set Register		4000E0A4
Interrupt Aggregator	0	GIRQ16 Result Register		4000E0A8
Interrupt Aggregator	0	GIRQ16 Enable Clear Register		4000E0AC
Interrupt Aggregator	0	GIRQ17 Source Register		4000E0B4
Interrupt Aggregator	0	GIRQ17 Enable Set Register		4000E0B8
Interrupt Aggregator	0	GIRQ17 Result Register		4000E0BC
Interrupt Aggregator	0	GIRQ17 Enable Clear Register		4000E0C0
Interrupt Aggregator	0	GIRQ18 Source Register		4000E0C8
Interrupt Aggregator	0	GIRQ18 Enable Set Register		4000E0CC
Interrupt Aggregator	0	GIRQ18 Result Register		4000E0D0
Interrupt Aggregator	0	GIRQ18 Enable Clear Register		4000E0D4

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Interrupt Aggregator	0	GIRQ19 Source Register		4000E0DC
Interrupt Aggregator	0	GIRQ19 Enable Set Register		4000E0E0
Interrupt Aggregator	0	GIRQ19 Result Register		4000E0E4
Interrupt Aggregator	0	GIRQ19 Enable Clear Register		4000E0E8
Interrupt Aggregator	0	GIRQ20 Source Register		4000E0F0
Interrupt Aggregator	0	GIRQ20 Enable Set Register		4000E0F4
Interrupt Aggregator	0	GIRQ20 Result Register		4000E0F8
Interrupt Aggregator	0	GIRQ20 Enable Clear Register		4000E0FC
Interrupt Aggregator	0	GIRQ21 Source Register		4000E104
Interrupt Aggregator	0	GIRQ21 Enable Set Register		4000E108
Interrupt Aggregator	0	GIRQ21 Result Register		4000E10C
Interrupt Aggregator	0	GIRQ21 Enable Clear Register		4000E110
Interrupt Aggregator	0	GIRQ22 Source Register		4000E118
Interrupt Aggregator	0	GIRQ22 Enable Set Register		4000E11C
Interrupt Aggregator	0	GIRQ22 Result Register		4000E120
Interrupt Aggregator	0	GIRQ22 Enable Clear Register		4000E124
Interrupt Aggregator	0	GIRQ23 Source Register		4000E12C
Interrupt Aggregator	0	GIRQ23 Enable Set Register		4000E130
Interrupt Aggregator	0	GIRQ23 Result Register		4000E134
Interrupt Aggregator	0	GIRQ23 Enable Clear Register		4000E138
Interrupt Aggregator	0	GIRQ24 Source Register		4000E140
Interrupt Aggregator	0	GIRQ24 Enable Set Register		4000E144
Interrupt Aggregator	0	GIRQ24 Result Register		4000E148
Interrupt Aggregator	0	GIRQ24 Enable Clear Register		4000E14C
Interrupt Aggregator	0	GIRQ25 Source Register		4000E154
Interrupt Aggregator	0	GIRQ25 Enable Set Register		4000E158
Interrupt Aggregator	0	GIRQ25 Result Register		4000E15C
Interrupt Aggregator	0	GIRQ25 Enable Clear Register		4000E160
Interrupt Aggregator	0	GIRQ26 Source Register		4000E168
Interrupt Aggregator	0	GIRQ26 Enable Set Register		4000E16C
Interrupt Aggregator	0	GIRQ26 Result Register		4000E170
Interrupt Aggregator	0	GIRQ26 Enable Clear Register		4000E174
Interrupt Aggregator	0	Block Enable Set Register		4000E200
Interrupt Aggregator	0	Block Enable Clear Register		4000E204
Interrupt Aggregator	0	Block IRQ Vector Register		4000E208
EC Register Bank	0	TEST		4000FC00
EC Register Bank	0	AHB Error Address Register		4000FC04
EC Register Bank	0	TEST		4000FC08
EC Register Bank	0	TEST		4000FC0C
EC Register Bank	0	TEST		4000FC10
EC Register Bank	0	AHB Error Control Register		4000FC14
EC Register Bank	0	Interrupt Control Register		4000FC18
EC Register Bank	0	ETM TRACE Enable Register		4000FC1C
EC Register Bank	0	Debug Enable Register		4000FC20

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
EC Register Bank	0	TEST		4000FC24
EC Register Bank	0	WDT Event Count Register		4000FC28
EC Register Bank	0	PECI DISABLE Register		4000FC40
EC Register Bank	0	TEST		4000FC44
EC Register Bank	0	TEST		4000FC48
EC Register Bank	0	TEST		4000FC4C
EC Register Bank	0	TEST		4000FC60
EC Register Bank	0	GPIO Bank Power Register		4000FC64
EC Register Bank	0	TEST		4000FC68
EC Register Bank	0	TEST		4000FC6C
EC Register Bank	0	Vwire FW Override Register		4000FC90
EC Register Bank	0	Other IP trim Register		4000FCF0
EC Register Bank	0	TEST		4000FD00
EC Register Bank	0	FW Scratch Register0		4000FD80
EC Register Bank	0	FW Scratch Register1		4000FD84
EC Register Bank	0	FW Scratch Register2		4000FD88
EC Register Bank	0	FW Scratch Register3		4000FD8C
Power Clocks and Resets	0	System Sleep Control Register		40080100
Power Clocks and Resets	0	Processor Clock Control Register		40080104
Power Clocks and Resets	0	Slow Clock Control Register		40080108
Power Clocks and Resets	0	Oscillator ID Register		4008010C
Power Clocks and Resets	0	PCR Power Reset Status Register		40080110
Power Clocks and Resets	0	Power Reset Control Register		40080114
Power Clocks and Resets	0	System Reset Register		40080118
Power Clocks and Resets	0	TEST		4008011C
Power Clocks and Resets	0	TEST		40080120
Power Clocks and Resets	0	Sleep Enable 0 Register		40080130
Power Clocks and Resets	0	Sleep Enable 1 Register		40080134
Power Clocks and Resets	0	Sleep Enable 2 Register		40080138
Power Clocks and Resets	0	Sleep Enable 3 Register		4008013C
Power Clocks and Resets	0	Sleep Enable 4 Register		40080140
Power Clocks and Resets	0	Clock Required 0 Register		40080150
Power Clocks and Resets	0	Clock Required 1 Register		40080154
Power Clocks and Resets	0	Clock Required 2 Register		40080158
Power Clocks and Resets	0	Clock Required 3 Register		4008015C
Power Clocks and Resets	0	Clock Required 4 Register		40080160
Power Clocks and Resets	0	Reset Enable 0 Register		40080170
Power Clocks and Resets	0	Reset Enable 1 Register		40080174
Power Clocks and Resets	0	Reset Enable 2 Register		40080178
Power Clocks and Resets	0	Reset Enable 3 Register		4008017C
Power Clocks and Resets	0	Reset Enable 4 Register		40080180
Power Clocks and Resets	0	Peripheral Reset Lock Register		40080184
GPIO	0	GPIO000 Pin Control Register		40081000
GPIO	0	GPIO002 Pin Control Register		40081008

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO003 Pin Control Register		4008100C
GPIO	0	GPIO004 Pin Control Register		40081010
GPIO	0	GPIO012 Pin Control Register		40081028
GPIO	0	GPIO013 Pin Control Register		4008102C
GPIO	0	GPIO015 Pin Control Register		40081034
GPIO	0	GPIO016 Pin Control Register		40081038
GPIO	0	GPIO020 Pin Control Register		40081040
GPIO	0	GPIO021 Pin Control Register		40081044
GPIO	0	GPIO026 Pin Control Register		40081058
GPIO	0	GPIO027 Pin Control Register		4008105C
GPIO	0	GPIO030 Pin Control Register		40081060
GPIO	0	GPIO032 Pin Control Register		40081068
GPIO	0	GPIO034 Pin Control Register		40081070
GPIO	0	GPIO045 Pin Control Register		40081094
GPIO	0	GPIO046 Pin Control Register		40081098
GPIO	0	GPIO047 Pin Control Register		4008109C
GPIO	0	GPIO050 Pin Control Register		400810A0
GPIO	0	GPIO051 Pin Control Register		400810A4
GPIO	0	GPIO053 Pin Control Register		400810AC
GPIO	0	GPIO055 Pin Control Register		400810B4
GPIO	0	GPIO056 Pin Control Register		400810B8
GPIO	0	GPIO057 Pin Control Register		400810BC
GPIO	0	GPIO063 Pin Control Register		400810CC
GPIO	0	GPIO070 Pin Control Register		400810E0
GPIO	0	GPIO071 Pin Control Register		400810E4
GPIO	0	GPIO104 Pin Control Register		40081110
GPIO	0	GPIO105 Pin Control Register		40081114
GPIO	0	GPIO106 Pin Control Register		40081118
GPIO	0	GPIO107 Pin Control Register		4008111C
GPIO	0	GPIO112 Pin Control Register		40081128
GPIO	0	GPIO113 Pin Control Register		4008112C
GPIO	0	GPIO120 Pin Control Register		40081140
GPIO	0	GPIO121 Pin Control Register		40081144
GPIO	0	GPIO122 Pin Control Register		40081148
GPIO	0	GPIO123 Pin Control Register		4008114C
GPIO	0	GPIO124 Pin Control Register		40081150
GPIO	0	GPIO125 Pin Control Register		40081154
GPIO	0	GPIO126 Pin Control Register		40081158
GPIO	0	GPIO127 Pin Control Register		4008115C
GPIO	0	GPIO130 Pin Control Register		40081160
GPIO	0	GPIO131 Pin Control Register		40081164
GPIO	0	GPIO132 Pin Control Register		40081168
GPIO	0	GPIO140 Pin Control Register		40081180
GPIO	0	GPIO143 Pin Control Register		4008118C

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO144 Pin Control Register		40081190
GPIO	0	GPIO145 Pin Control Register		40081194
GPIO	0	GPIO146 Pin Control Register		40081198
GPIO	0	GPIO147 Pin Control Register		4008119C
GPIO	0	GPIO150 Pin Control Register		400811A0
GPIO	0	GPIO156 Pin Control Register		400811B8
GPIO	0	GPIO157 Pin Control Register		400811BC
GPIO	0	GPIO163 Pin Control Register		400811CC
GPIO	0	GPIO165 Pin Control Register		400811D4
GPIO	0	GPIO170 Pin Control Register		400811E0
GPIO	0	GPIO171 Pin Control Register		400811E4
GPIO	0	GPIO200 Pin Control Register		40081200
GPIO	0	GPIO201 Pin Control Register		40081204
GPIO	0	GPIO202 Pin Control Register		40081208
GPIO	0	GPIO203 Pin Control Register		4008120C
GPIO	0	GPIO204 Pin Control Register		40081210
GPIO	0	GPIO223 Pin Control Register		4008124C
GPIO	0	GPIO224 Pin Control Register		40081250
GPIO	0	GPIO227 Pin Control Register		4008125C
GPIO	0	GPIO250 Pin Control Register		400812A0
GPIO	0	GPIO253 Pin Control Register		400812AC
GPIO	0	Input GPIO[000:036]		40081300
GPIO	0	Input GPIO[040:076]		40081304
GPIO	0	Input GPIO[100:127]		40081308
GPIO	0	Input GPIO[140:176]		4008130C
GPIO	0	Input GPIO[200:236]		40081310
GPIO	0	Input GPIO[240:276]		40081314
GPIO	0	Output GPIO[000:036]		40081380
GPIO	0	Output GPIO[040:076]		40081384
GPIO	0	Output GPIO[100:127]		40081388
GPIO	0	Output GPIO[140:176]		4008138C
GPIO	0	Output GPIO[200:236]		40081390
GPIO	0	Output GPIO[240:276]		40081394
GPIO	0	GPIO000 Pin Control2 Register		40081500
GPIO	0	GPIO002 Pin Control2 Register		40081508
GPIO	0	GPIO003 Pin Control2 Register		4008150C
GPIO	0	GPIO004 Pin Control2 Register		40081510
GPIO	0	GPIO012 Pin Control2 Register		40081528
GPIO	0	GPIO013 Pin Control2 Register		4008152C
GPIO	0	GPIO015 Pin Control2 Register		40081534
GPIO	0	GPIO016 Pin Control2 Register		40081538
GPIO	0	GPIO020 Pin Control2 Register		40081540
GPIO	0	GPIO021 Pin Control2 Register		40081544
GPIO	0	GPIO026 Pin Control2 Register		40081558

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO027 Pin Control2 Register		4008155C
GPIO	0	GPIO030 Pin Control2 Register		40081560
GPIO	0	GPIO032 Pin Control2 Register		40081568
GPIO	0	GPIO034 Pin Control2 Register		40081570
GPIO	0	GPIO045 Pin Control2 Register		40081594
GPIO	0	GPIO046 Pin Control2 Register		40081598
GPIO	0	GPIO047 Pin Control2 Register		4008159C
GPIO	0	GPIO050 Pin Control2 Register		400815A0
GPIO	0	GPIO051 Pin Control2 Register		400815A4
GPIO	0	GPIO053 Pin Control2 Register		400815AC
GPIO	0	GPIO055 Pin Control2 Register		400815B4
GPIO	0	GPIO056 Pin Control2 Register		400815B8
GPIO	0	GPIO057 Pin Control2 Register		400815BC
GPIO	0	GPIO063 Pin Control2 Register		400815CC
GPIO	0	GPIO070 Pin Control2 Register		400815E0
GPIO	0	GPIO071 Pin Control2 Register		400815E4
GPIO	0	GPIO104 Pin Control2 Register		40081610
GPIO	0	GPIO105 Pin Control2 Register		40081614
GPIO	0	GPIO106 Pin Control2 Register		40081618
GPIO	0	GPIO107 Pin Control2 Register		4008161C
GPIO	0	GPIO112 Pin Control2 Register		40081628
GPIO	0	GPIO113 Pin Control2 Register		4008162C
GPIO	0	GPIO120 Pin Control2 Register		40081640
GPIO	0	GPIO121 Pin Control2 Register		40081644
GPIO	0	GPIO122 Pin Control2 Register		40081648
GPIO	0	GPIO123 Pin Control2 Register		4008164C
GPIO	0	GPIO124 Pin Control2 Register		40081650
GPIO	0	GPIO125 Pin Control2 Register		40081654
GPIO	0	GPIO126 Pin Control2 Register		40081658
GPIO	0	GPIO127 Pin Control2 Register		4008165C
GPIO	0	GPIO130 Pin Control2 Register		40081660
GPIO	0	GPIO131 Pin Control2 Register		40081664
GPIO	0	GPIO132 Pin Control2 Register		40081668
GPIO	0	GPIO140 Pin Control2 Register		40081680
GPIO	0	GPIO143 Pin Control2 Register		4008168C
GPIO	0	GPIO144 Pin Control2 Register		40081690
GPIO	0	GPIO145 Pin Control2 Register		40081694
GPIO	0	GPIO146 Pin Control2 Register		40081698
GPIO	0	GPIO147 Pin Control2 Register		4008169C
GPIO	0	GPIO150 Pin Control2 Register		400816A0
GPIO	0	GPIO156 Pin Control2 Register		400816B8
GPIO	0	GPIO157 Pin Control2 Register		400816BC
GPIO	0	GPIO163 Pin Control2 Register		400816CC
GPIO	0	GPIO165 Pin Control2 Register		400816D4

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO170 Pin Control2 Register		400816E0
GPIO	0	GPIO171 Pin Control2 Register		400816E4
GPIO	0	GPIO200 Pin Control2 Register		40081700
GPIO	0	GPIO201 Pin Control2 Register		40081704
GPIO	0	GPIO202 Pin Control2 Register		40081708
GPIO	0	GPIO203 Pin Control2 Register		4008170C
GPIO	0	GPIO204 Pin Control2 Register		40081710
GPIO	0	GPIO223 Pin Control2 Register		4008174C
GPIO	0	GPIO224 Pin Control2 Register		40081750
GPIO	0	GPIO227 Pin Control2 Register		4008175C
GPIO	0	GPIO250 Pin Control2 Register		400817A0
GPIO	0	GPIO253 Pin Control2 Register		400817AC
UART	0	Receive Buffer Register	Run-time	400F2400
UART	0	Transmit Buffer Register	Run-time	400F2400
UART	0	Programmable Baud Rate Generator LSB Register	Run-time	400F2400
UART	0	Programmable Baud Rate Generator MSB Register	Run-time	400F2401
UART	0	Interrupt Enable Register	Run-time	400F2401
UART	0	FIFO Control Register	Run-time	400F2402
UART	0	Interrupt Identification Register	Run-time	400F2402
UART	0	Line Control Register	Run-time	400F2403
UART	0	Modem Control Register	Run-time	400F2404
UART	0	Line Status Register	Run-time	400F2405
UART	0	Modem Status Register	Run-time	400F2406
UART	0	Scratchpad Register	Run-time	400F2407
UART	0	Activate Register	Config	400F2730
UART	0	Configuration Select Register	Config	400F27F0
UART	1	Receive Buffer Register	Run-time	400F2800
UART	1	Transmit Buffer Register	Run-time	400F2800
UART	1	Programmable Baud Rate Generator LSB Register	Run-time	400F2800

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
UART	1	Programmable Baud Rate Generator MSB Register	Run-time	400F2801
UART	1	Interrupt Enable Register	Run-time	400F2801
UART	1	FIFO Control Register	Run-time	400F2802
UART	1	Interrupt Identification Register	Run-time	400F2802
UART	1	Line Control Register	Run-time	400F2803
UART	1	Modem Control Register	Run-time	400F2804
UART	1	Line Status Register	Run-time	400F2805
UART	1	Modem Status Register	Run-time	400F2806
UART	1	Scratchpad Register	Run-time	400F2807
UART	1	Activate Register	Config	400F2B30
UART	1	Configuration Select Register	Config	400F2BF0
UART	2	Receive Buffer Register	Run-time	400F2C00
UART	2	Transmit Buffer Register	Run-time	400F2C00
UART	2	Programmable Baud Rate Generator LSB Register	Run-time	400F2C00
UART	2	Programmable Baud Rate Generator MSB Register	Run-time	400F2C01
UART	2	Interrupt Enable Register	Run-time	400F2C01
UART	2	FIFO Control Register	Run-time	400F2C02
UART	2	Interrupt Identification Register	Run-time	400F2C02
UART	2	Line Control Register	Run-time	400F2C03
UART	2	Modem Control Register	Run-time	400F2C04
UART	2	Line Status Register	Run-time	400F2C05
UART	2	Modem Status Register	Run-time	400F2C06
UART	2	Scratchpad Register	Run-time	400F2C07
UART	2	Activate Register	Config	400F2F30

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
UART	2	Configuration Select Register	Config	400F2FF0
Real Time Clock	0	Seconds Register	Runtime	400F5000
Real Time Clock	0	Seconds Alarm Register	Runtime	400F5001
Real Time Clock	0	Minutes Register	Runtime	400F5002
Real Time Clock	0	Minutes Alarm Register	Runtime	400F5003
Real Time Clock	0	Hours Register	Runtime	400F5004
Real Time Clock	0	Hours Alarm Register	Runtime	400F5005
Real Time Clock	0	Day of Week Register	Runtime	400F5006
Real Time Clock	0	Day of Month Register	Runtime	400F5007
Real Time Clock	0	Month Register	Runtime	400F5008
Real Time Clock	0	Year Register	Runtime	400F5009
Real Time Clock	0	Register A	Runtime	400F500A
Real Time Clock	0	Register B	Runtime	400F500B
Real Time Clock	0	Register C	Runtime	400F500C
Real Time Clock	0	Register D	Runtime	400F500D
Real Time Clock	0	Reserved	Runtime	400F500E
Real Time Clock	0	Reserved	Runtime	400F500F
Real Time Clock	0	RTC Control Register	Runtime	400F5010
Real Time Clock	0	Week Alarm Register	Runtime	400F5014
Real Time Clock	0	Daylight Savings Forward Register	Runtime	400F5018
Real Time Clock	0	Daylight Savings Backward Register	Runtime	400F501C
Real Time Clock	0	TEST	Runtime	400F5020
Global Configuration	0	Global Configuration Reserved	Runtime	400FFF00
Global Configuration	0	Control	Runtime	400FFF02

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Global Configuration	0	Logical Device Number	Run-time	400FFF07
Global Configuration	0	Device Revision	Run-time	400FFF1C
Global Configuration	0	Device Sub ID	Run-time	400FFF1D
Global Configuration	0	Device ID[7:0]	Run-time	400FFF1E
Global Configuration	0	Device ID[15:0]	Run-time	400FFF1F
Global Configuration	0	Legacy Device ID	Run-time	400FFF20
Global Configuration	0	TEST	Run-time	400FFF28
Global Configuration	0	TEST	Run-time	400FFF29
Global Configuration	0	Test0	Run-time	400FFF2A
Global Configuration	0	Test1	Run-time	400FFF2B
Global Configuration	0	TEST	Run-time	400FFF2C
Global Configuration	0	TEST	Run-time	400FFF2D
Global Configuration	0	TEST	Run-time	400FFF2E
Global Configuration	0	TEST	Run-time	400FFF2F
ARM M4	0	Auxiliary_Control		E000E008
ARM M4	0	SystemTick_Ctrl_Status		E000E010
ARM M4	0	SystemTick_Reload_Value		E000E014
ARM M4	0	SystemTick_Current_Value		E000E018
ARM M4	0	SystemTick_Calibration_Value		E000E01C
ARM M4	0	CPU_ID		E000ED00
ARM M4	0	Interrupt_Ctl_and_State		E000ED04
ARM M4	0	Vector_Table_Offset		E000ED08
ARM M4	0	Application_Interrupt_and_Reset_Ctl		E000ED0C
ARM M4	0	System_Ctl		E000ED10
ARM M4	0	Config_and_Ctl		E000ED14
ARM M4	0	System_Handler_Priority1		E000ED18
ARM M4	0	System_Handler_Priority2		E000ED1C
ARM M4	0	System_Handler_Priority3		E000ED20
ARM M4	0	System_Handler_Ctl_and_State		E000ED24
ARM M4	0	Configurable_Fault_Status		E000ED28
ARM M4	0	Hard_Fault_Status		E000ED2C
ARM M4	0	Debug_Fault_Status		E000ED30
ARM M4	0	Debug_Halting_Ctl_and_Status		E000EDF0

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ARM M4	0	Debug_Core_Register_Selector		E000EDF4
ARM M4	0	Debug_Core_Register_Data		E000EDF8
ARM M4	0	Debug_Exception_and_Monitor_Ctl		E000EDFC
ARM M4	0	Bus_Fault_Address		E000ED38
ARM M4	0	Auxiliary_Fault_Status		E000ED3C
ARM M4	0	Processor_Feature0		E000ED40
ARM M4	0	Processor_Feature1		E000ED44
ARM M4	0	Debug_Features0		E000ED48
ARM M4	0	Auxiliary_Features0		E000ED4C
ARM M4	0	Memory_Model_Feature0		E000ED50
ARM M4	0	Memory_Model_Feature1		E000ED54
ARM M4	0	Memory_Model_Feature2		E000ED58
ARM M4	0	Memory_Model_Feature3		E000ED5C
ARM M4	0	Instruction_Set_Attributes0		E000ED60
ARM M4	0	Instruction_Set_Attributes1		E000ED64
ARM M4	0	Instruction_Set_Attributes2		E000ED68
ARM M4	0	Instruction_Set_Attributes3		E000ED6C
ARM M4	0	Instruction_Set_Attributes4		E000ED70
ARM M4	0	Coprocessor_Access_Ctl		E000ED88
ARM M4	0	Software_Triggered_Interrupt		E000EF00

4.0 POWER, CLOCKS, AND RESETS

4.1 Introduction

The Power, Clocks, and Resets (PCR) chapter identifies all the power supplies, clock sources, and reset inputs to the chip and defines all the derived power, clock, and reset signals. In addition, this section identifies Power, Clock, and Reset events that may be used to generate an interrupt event, as well as, the Chip Power Management Features.

4.2 References

No references have been cited for this chapter.

4.3 Interrupts

The Power, Clocks, and Resets logic generates no events

4.4 Power

TABLE 4-1: POWER SOURCE DEFINITIONS

Power Well	Nominal Voltage	Description	Source
VTR_REG	1.8V - 3.3V	This supply is used to derive the chip's core power.	Pin Interface
VTR_ANALOG	3.3V	3.3V Analog Power Supply.	
VTR_PLL	3.3V	3.3V Power Supply for the 48MHz PLL. This must be connected to the same supply as VTR_ANALOG.	Pin Interface
VTR1	3.3V	3.3V System Power Supply. This is typically connected to the “Always-on” or “Suspend” supply rails in system. This supply must be on prior to the system RSMRST# signal being deasserted	Pin Interface
VTR2	3.3V or 1.8V	3.3V or 1.8V System Power Supply. This supply is used to power one bank of I/O pins. See Note 1.	Pin Interface
VTR_CORE	1.2V	The main power well for internal logic	Internal regulator
VBAT	3.0V - 3.3V	System Battery Back-up Power Well. This is the “coin-cell” battery. GPIOs that share pins with VBAT signals are powered by this supply.	Pin Interface VBAT
VSS	0V	Digital Ground	Pin Interface

Note 1: See Section 4.4.1, "I/O Rail Requirements" for connection requirements for VTRx.

2: The source for the Internal regulator is VTR_REG.

3: VTR refers to VTR_REG and VTR_ANALOG.

4.4.1 I/O RAIL REQUIREMENTS

All pins are powered by the power supply pins: VBAT, VTR1, VTR2. The VBAT supply must be 3V to 3.6V maximum, as shown in the following section. The VTR1 is fixed 3.3V and VTR2 pins may be connected to either a 3.3V or a 1.8V power supply as configured by the firmware.

If a power rail is not powered and stable when RESET_SYS is de-asserted and is not required for booting, software can configure the pins on that bank appropriately by setting the corresponding bit in the GPIO Bank Power Register, once software can determine that the power supply is up and stable. All GPIOs in the bank must be left in their default state and not modified until the Bank Power is configured properly.

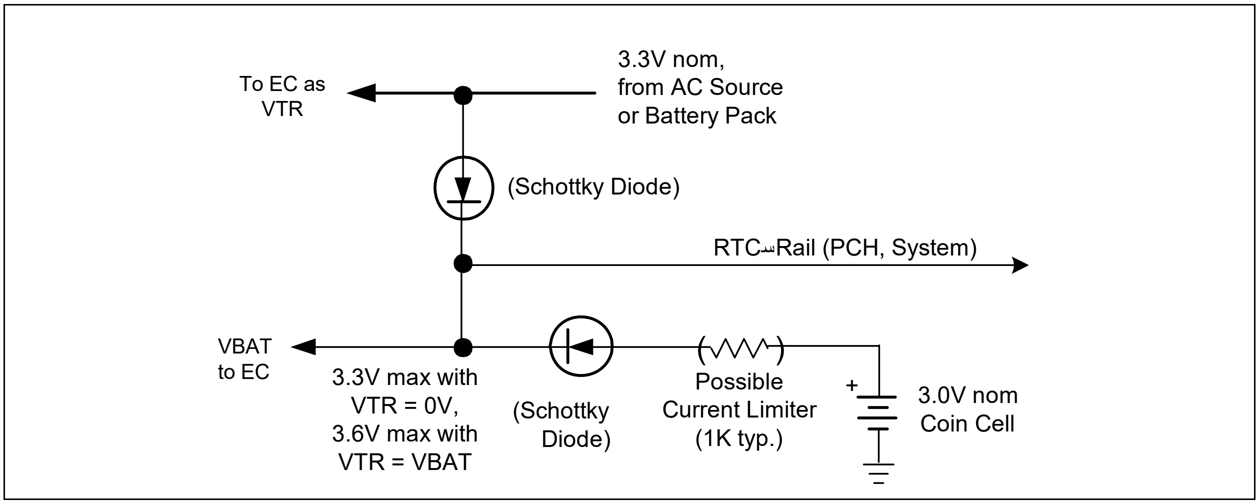
4.4.2 BATTERY CIRCUIT REQUIREMENTS

VBAT must always be present if VTR_ANALOG is present.

Microchip recommends removing all power sources to the device defined in Table 4-1, "Power Source Definitions" and all external voltage references defined in Table 4-2, "Voltage Reference Definitions" before removing and replacing the battery. In addition, upon removing the battery, discharge the battery pin before replacing the battery.

The following external circuit is recommended to fulfill this requirement:

FIGURE 4-1: RECOMMENDED BATTERY CIRCUIT



4.4.3 VOLTAGE REFERENCES

Table 4-2 lists the External Voltage References to which the CEC1712 provides high impedance interfaces.

TABLE 4-2: VOLTAGE REFERENCE DEFINITIONS

Power Well	Nominal Input Voltage	Scaling Ratio	Nominal Monitored Voltage	Description	Source
VREF_VTT	Variable	n/a	Variable	Processor Voltage External Voltage Reference Used to scale Processor Interface signals. (See Note)	Pin Interface
VREF_ADC	Variable	n/a	Variable	ADC Reference Voltage	Pin Interface
Note: In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_VTT Disable bit to 1. This bit is defined in PECI Disable Register bit 0.					

4.4.4 SYSTEM POWER SEQUENCING

The following table defines the behavior of the main power rails in each of the defined ACPI power states.

TABLE 4-3: TYPICAL POWER SUPPLIES VS. ACPI POWER STATES

Supply Name	ACPI Power State						Description
	S0 (FULL ON)	S1 (POS)	S3 (STR)	S4 (STD)	S5 (Soft Off)	G3 (MECH Off)	
VTR1	ON	ON	ON	ON	ON	OFF	“Always-on” Supply
VTR2	ON	ON	ON	ON	ON	OFF	3.3V/1.8V Power Supply for Bank 2
VBAT	ON	ON	ON	ON Note	ON Note	ON Note	Battery Back-up Supply

Note: This device requires that the VBAT power is on when the VTR(Note 3) power supply is on. External circuitry, a diode isolation circuit, is implemented on the motherboard to extend the battery life. This external circuitry ensures the VBAT pin will derive power from the VTR power well when it is on. Therefore, the VBAT supply will never appear to be off when the VTR rail is on.

4.5 Clocks

The following section defines the clocks that are generated and derived.

4.5.1 RAW CLOCK SOURCES

The table defines raw clocks .

TABLE 4-4: SOURCE CLOCK DEFINITIONS

Clock Name	Frequency	Description	Source
32KHZ_IN	32.768 kHz (nominal)	Single-ended external clock input pin	32KHZ_IN pin
32.768 kHz Crystal Oscillator	32.768 kHz	A 32.768 kHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins. The accuracy of the clock depends on the accuracy of the crystal and the characteristics of the analog components used as part of the oscillator The crystal oscillator source can bypass the crystal with a single-ended clock input. This option is configured with the Clock Enable Register.	Pin Interface (XTAL1 and XTAL2) When used singled-ended, input pin is XTAL2
32.768 kHz Silicon Oscillator	32.768 kHz	32.768 kHz low power Internal Oscillator. The frequency is 32.768KHz ±2%.	Internal Oscillator powered by VBAT.
32 MHz Ring Oscillator	32MHz	The 32MHz Ring Oscillator is used to supply a clock for the 48MHz main clock domain while the 48MHz PLL is not locked. Its frequency can range from 32Mhz to 96MHz. This clock is passed though a divide by 2 circuit to generate clock in the range 16MHz to 48MHz	Powered by VTR_CORE.

TABLE 4-4: SOURCE CLOCK DEFINITIONS (CONTINUED)

Clock Name	Frequency	Description	Source
48 MHz PLL	48MHz	The 48 MHz Phase Locked Loop generates a 48MHz clock locked to the VBAT 32KHz Clock	Powered by VTR_CORE. May be stopped by Chip Power Management Features.

4.5.2 CLOCK DOMAINS

TABLE 4-5: CLOCK DOMAIN DEFINITIONS

Clock Domain	Description
VBAT 32KHz Clock	The clock source used as reference for PLL lock and System Clock controls.
32KHz	The clock source used by internal blocks that require an always-on low speed clock
48MHz	The main clock source used by most internal blocks
100KHz	A low-speed clock derived from the 48MHz clock domain. Used as a time base for PWMs and Tachs.
EC_CLK	The clock used by the EC processor. The frequency is determined by the Processor Clock Control Register.

4.5.3 48MHZ PLL

The 48MHz clock domain is primarily driven by a 48MHz PLL, which derives 48MHz from the VBAT 32KHz Clock domain. In Heavy Sleep mode, the 48MHz PLL is shut off. When the PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset, the 32MHz ring oscillator becomes the clock source for the 48MHz clock domain until the PLL is stable. The PLL becomes stable after about 3ms after the VBAT 32KHz Clock is stable; until that time, the 48MHz clock domain may range from 16MHz to 48MHz, as this is the accuracy range of the 32MHz ring.

The PLL requires its own power 3.3V power supply, VTR_PLL. This power rail must be active and stable no later than the latest of VTR_REG and VTR_ANALOG. There is no hardware detection of VTR_PLL power good in the reset generator.

4.5.4 32KHZ CLOCK

The 32kHz Clock Domain may be sourced from a crystal oscillator, using an external crystal, by an internal 32kHz oscillator, or from a single-ended clock input. The external single-ended clock source can itself be sourced from the 32KHZ_IN signal that is a GPIO alternate function or from the XTAL2 crystal pin. The Clock Enable Register is used to configure the source for the 32 kHz clock domain.

When VTR_CORE is off, the 32 kHz clock domain can be disabled, for lowest standby power, or it can be kept running in order to provide a clock for the Real Time Clock or the Week Timer.

An external single-ended clock input for 32KHZ_IN may be supplied by any accurate 32KHz clock source in the system. The SUSCLK output from the chipset may be used as the 32KHz source. SUSCLK must be present when VTR is on. See chipset documentation for details on the use of SUSCLK.

If firmware switches the 32KHz clock source, the 48MHz PLL will be shut off and then restarted. The 48MHz clock domain will become unlocked and be sourced from the 32 MHz Ring Oscillator until the 48MHz PLL is on and locked.

4.5.4.1 VBAT 32KHz Clock

This clock source is used to drive the 48MHz PLL. VBAT 32KHz Clock should remain on while the 48Mhz PLL is ON. The internal source provides a reference for the Activity Detect that monitors the external clock input, as well as providing a low latency backup clock source when the Activity Detector cannot detect a clock on the external input.

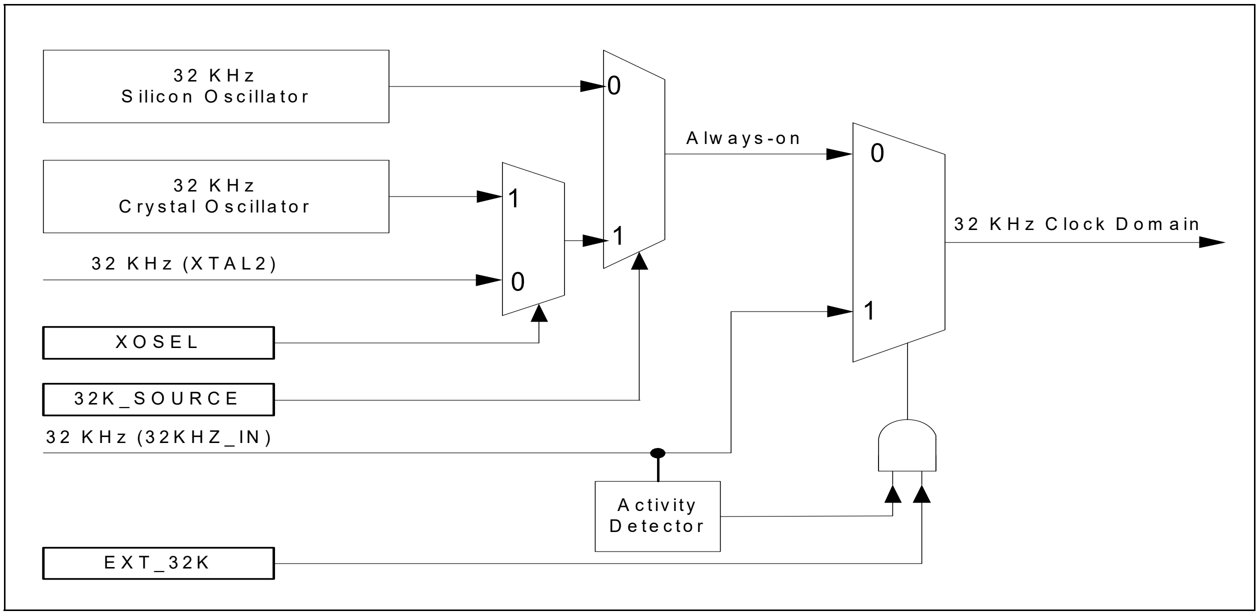
The VBAT 32KHz Clock Internal Clock Source can be driven either by the 32.768 kHz Silicon Oscillator or the 32.768 kHz Crystal Oscillator.

4.5.4.2 External 32KHz Clock Activity Detector

When the EXT_32K field in the Clock Enable Register is set for an external clock source an Activity Detector monitors the external 32KHz signal at all times. If there is no clock detected on the pin, the 32KHZ clock domain is switched to the internal 32KHz silicon oscillator. If a clock is again detected on the pin, the 32KHz clock domain is switched to the pin

The following figure illustrates the 32KHz clock domain sourcing.

FIGURE 4-2: 32KHZ ACTIVITY DETECTOR



4.5.4.3 32KHz Crystal Oscillator

If the 32KHz source will never be the crystal oscillator, then the XTAL2 pin should be grounded. The XTAL1 pin should be left unconnected.

4.6 Resets

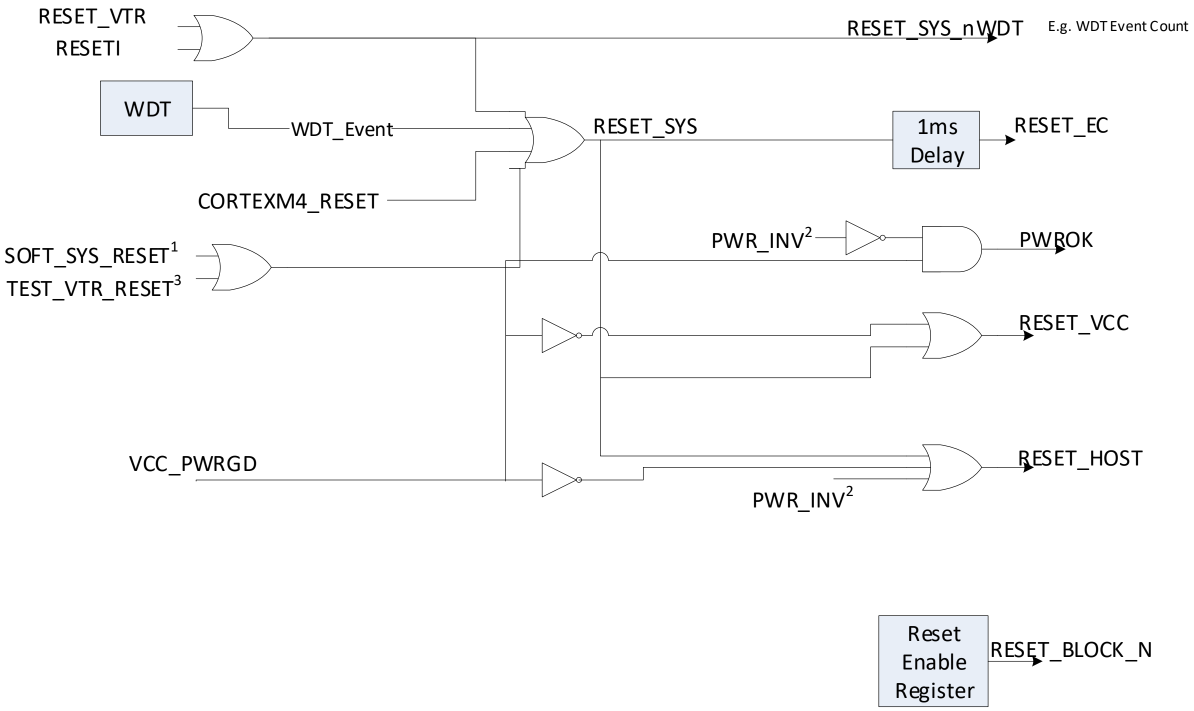
TABLE 4-6: DEFINITION OF RESET SIGNALS

Reset	Description	Source
RESET_VBAT	Internal VBAT Reset signal. This signal is used to reset VBAT powered registers.	RESET_VBAT is a pulse that is asserted at the rising edge of VTR power if the VBAT voltage is below a nominal 1.25V. RESET_VBAT is also asserted as a level if, while VTR power is not present, the coin cell is replaced with a new cell that delivers at least a nominal 1.25V. In this latter case RESET_VBAT is de-asserted when VTR power is applied. No action is taken if the coin cell is replaced, or if the VBAT voltage falls below 1.25 V nominal, while VTR power is present.
RESET_VTR	Internal VTR Reset signal.	This internal reset signal is asserted as long as the reset generator determines that the output of the internal regulator is stable at its target voltage and that the voltage rail supplying the main clock PLL is at 3.3V. Although most VTR_CORE-powered registers are reset on RESET_SYS, some registers are only reset on this reset.

TABLE 4-6: DEFINITION OF RESET SIGNALS (CONTINUED)

Reset	Description	Source
RESET_SYS	Internal Reset signal. This signal is used to reset VTR_CORE powered registers.	RESET_SYS is the main global reset signal. This reset signal will be asserted if: • RESET_VTR is asserted • The nRESET_IN pin asserted • A WDT Event event is asserted • A soft reset is asserted by the SOFT_SYS-_RESET bit in the System Reset Register • ARM M4 SYSRESETREQ
RESET_VCC	Performs a reset when Host power (VCC) is turned off	This signal is asserted if **Note:** RESET_SYS is asserted
RESET_HOST	Performs a reset when VCC_PWRGD is low	This signal is asserted if • RESET_SYS is asserted • VCC_PWRGD is low • The PWR_INV bit in the Power Reset Control Register is ‘1b’1
WDT Event	A WDT Event generates the RESET_SYS event. This signal resets VTR_CORE powered registers with the exception of the WDT Event Count Register register. Note that the glitch protect circuits do not activate on a WDT reset. WDT Event does not reset VBAT registers or logic.	This reset signal will be asserted if: • A WDT Event event is asserted This event is indicated by the WDT bit in the Power-Fail and Reset Status Register
RESET_SYS_nWDT	Internal Reset signal. This signal is used to reset VTR_CORE powered registers not effected by a WDT Event A RESET_SYS_nWDT is used to reset registers that need to be preserved through a WDT Event like a WDT Event Count Register.	This reset signal will be asserted if: • RESET_VTR is asserted • The nRESET_IN pin asserted
RESET_EC	Internal reset signal to reset the processor in the EC Subsystem.	This reset is a stretched version of RESET_SYS. This reset asserts at the same time that RESET_SYS asserts and is held asserted for 1ms after RESET_SYS deasserts.
RESET_BLOCK_N	Each IP block in the device may be configured to be reset by setting the RESET_ENABLE register.	This reset signal will be asserted if Block N RESET_ENABLE is set to 1 and Peripheral Reset Enable n Register is unlocked.

FIGURE 4-3: RESETS BLOCK DIAGRAM



Note 1: SOFT_SYS_RESET is implemented in bit[8] of the System Reset Register
Note 2: PWR_INV is implemented in bit[0] of the Power Reset Control Register
Note 3: TEST_VTR_RESET is implemented in bit [1] of the Host Global Test Register

4.7 Chip Power Management Features

This device is designed to always operate in its lowest power state during normal operation. In addition, this device offers additional programmable options to put individual logical blocks to sleep as defined in the following section, Section 4.7.1.

4.7.1 BLOCK LOW POWER MODES

All power related control signals are generated and monitored centrally in the chip's Power, Clocks, and Resets (PCR) block. The power manager of the PCR block uses a sleep interface to communicate with all the blocks. The sleep interface consists of three signals:

- **SLEEP_ENABLE (request to sleep the block)** is generated by the PCR block. A group of SLEEP_ENABLE signals are generated for every clock segment. Each group consists of a SLEEP_ENABLE signal for every block in that clock segment.
- **CLOCK_REQUIRED (request clock on)** is generated by every block. They are grouped by blocks on the same clock segment. The PCR monitors these signals to see when it can gate off clocks.

A block can always drive CLOCK_REQUIRED low synchronously, but it *must* drive it high asynchronously since its internal clocks are gated and it has to assume that the clock input itself is gated. Therefore the block can only drive CLOCK_REQUIRED high as a result of a register access or some other input signal.

The following table defines a block's power management protocol:

TABLE 4-7: POWER MANAGEMENT PROTOCOL

Power State	SLEEP_ENABLE	CLOCK_REQUIRED	Description
Normal operation	Low	Low	Block is idle and NOT requesting clocks. The block gates its own internal clock.
Normal operation	Low	High	Block is NOT idle and requests clocks.
Request sleep	Rising Edge	Low	Block is IDLE and enters sleep mode immediately. The block gates its own internal clock. The block cannot request clocks again until SLEEP_ENABLE goes low.
Request sleep	Rising Edge	High then Low	Block is not IDLE and will stop requesting clocks and enter sleep when it finishes what it is doing. This delay is block specific, but should be less than 1 ms. The block gates its own internal clock. After driving CLOCK_REQUIRED low, the block cannot request clocks again until SLEEP_ENABLE goes low.
Register Access	X	High	Register access to a block is always available regardless of SLEEP_ENABLE. Therefore the block ungates its internal clock and drives CLOCK_REQUIRED high during the access. The block will regate its internal clock and drive CLOCK_REQUIRED low when the access is done.

A wake event clears all SLEEP_ENABLE bits momentarily, and then returns the SLEEP_ENABLE bits back to their original state. The block that needs to respond to the wake event will do so.

The Sleep Enable, Clock Required and Reset Enable Registers are defined in Section 4.8.

4.7.2 CONFIGURING THE CHIP'S SLEEP STATES

The chip supports two sleep states: LIGHT SLEEP and HEAVY SLEEP. The chip will enter one of these two sleep states only when all the blocks have been commanded to sleep and none of them require a 48MHz clock source (i.e., all CLOCK_REQUIRED status bits are 0), and the processor has executed its sleep instruction. These sleep states must be selected by firmware via the System Sleep Control bits implemented in the System Sleep Control Register prior to issuing the sleep instruction. Table 4-9, "System Sleep Modes" defines each of these sleep states.

There are two ways to command the chip blocks to enter sleep.

1. Assert the SLEEP_ALL bit located in the System Sleep Control Register
2. Assert all the individual block sleep enable bits

Blocks will only enter sleep after their sleep signal is asserted and they no longer require the 48MHz source. Each block has a corresponding clock required status bit indicating when the block has entered sleep. The general operation is that a block will keep the 48MHz clock source on until it completes its current transaction. Once the block has completed its work, it deasserts its clock required signal. Blocks like timers, PWMs, etc. will de-assert their clock required signals immediately. See the individual block Low Power Mode sections to determine how each individual block enters sleep.

4.7.3 DETERMINING WHEN THE CHIP IS SLEEPING

The TST_CLK_OUT pin can be used to verify the chip's clock has stopped, which indicates the device is in LIGHT SLEEP or HEAVY SLEEP, as determined by the System Sleep Control Register. If the clock is toggling the chip is in the full on running state. if the clock is not toggling the chip has entered the programmed sleep state.

4.7.4 WAKING THE CHIP FROM SLEEPING STATE

The chip will remain in the configured sleep state until it detects either a wake event or a full VTR_CORE POR. A wake event occurs when a wake-capable interrupt is enabled and triggered. Interrupts that are not wake-capable cannot occur while the system is in LIGHT SLEEP or HEAVY SLEEP.

In LIGHT SLEEP, the 48MHz clock domain is gated off, but the 48 MHz PLL remains operational and locked to the 32KHz clock domain. On wake, the PLL output is ungated and the 48MHz clock domain starts immediately, with the PLL_LOCK bit in the Oscillator ID Register set to '1'. Any device that requires an accurate clock, such as a UART, may be used immediately on wake.

In HEAVY SLEEP, the 48 MHz PLL is shut down. On wake, the 32 MHz Ring Oscillator is used to provide a clock source for the 48MHz clock domain until the PLL locks to the 32KHz clock domain. The ring oscillator starts immediately on wake, so there is no latency for the EC to start after a wake, However, the ring oscillator is only accurate to ±50%, so any device that requires an accurate 48MHz clock will not operate correctly until the PLL locks.The time to lock latency for the PLL is shown in Table 4-9, "System Sleep Modes".

The SLEEP_ALL bit is automatically cleared when the processor responds to an interrupt. This applies to non-wake interrupts as well as wake interrupts, in the event an interrupt occurs between the time the processor issued a WAIT FOR INTERRUPT instruction and the time the system completely enters the sleep state.

Any JTAG access to the ARM/STAP will cause a pseudo-wake event where the clocks are turned on, but the CHip is still in sleep (SLEEP_EN's and SLEEP_ALL stay in the same state).This way the access can occur over JTAG, without changing the parts state, and the part can go back to sleep once the JTAG access is over.

4.7.4.1 Wake-Only Events

Some devices which respond to an external master require the 48MHz clock domain to operate but do not necessarily require and immediate processing by the EC. Wake-only events provide the means to start the 48MHz clock domain without triggering an EC interrupt service routine. This events are grouped into a single GIRQ, GIRQ22. Events that are enabled in that GIRQ will start the clock domain when the event occurs, but will not invoke an EC interrupt. The SLEEP_ENABLE flags all remain asserted. If the activity for the event does not in turn trigger another EC interrupt, the CLOCK_REQUIRED for the block will re-assert and the configured sleep state will be re-entered.

4.8 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the Power, Clocks, and Resets Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 4-8: REGISTER SUMMARY

Offset	Name
0h	System Sleep Control Register
4h	Processor Clock Control Register
8h	Slow Clock Control Register
Ch	Oscillator ID Register
10h	PCR Power Reset Status Register
14h	Power Reset Control Register
18h	System Reset Register
1Ch	Reserved
20h	TEST
30h	Sleep Enable 0 Register
34h	Sleep Enable 1 Register
38h	Sleep Enable 2 Register
3Ch	Sleep Enable 3 Register
40h	Sleep Enable 4 Register
50h	Clock Required 0 Register
54h	Clock Required 1 Register
58h	Clock Required 2 Register
5Ch	Clock Required 3 Register
60h	Clock Required 4 Register
70h	Reset Enable 0 Register
74h	Reset Enable 1 Register
78h	Reset Enable 2 Register
7Ch	Reset Enable 3 Register
80h	Reset Enable 4 Register
84h	Peripheral Reset Lock Register

All register addresses are naturally aligned on 32-bit boundaries. Offsets for registers that are smaller than 32 bits are reserved and must not be used for any other purpose.

The bit definitions for the Sleep Enable, Clock Required and Reset Enable Registers are defined in the Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory".

4.9 Sleep Enable *n* Registers

4.9.1 SLEEP ENABLE *N* REGISTER

Offset	See Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	SLEEP_ENABLE 1=Block is commanded to sleep at next available moment 0=Block is free to use clocks as necessary Unassigned bits are reserved. They must be set to ‘1b’ when written. When read, unassigned bits return the last value written.	R/W	0h	RESET _SYS

4.9.2 CLOCK REQUIRED *N* REGISTER

Offset	See Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CLOCK_REQUIRED 1=Bock requires clocks 0=Block does not require clocks Unassigned bits are reserved and always return 0 when read.	R	0h	RESET _SYS

4.9.3 PERIPHERAL RESET ENABLE *N* REGISTER

Offset	See Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PERIPHERAL_RESET_ENABLE 1= Will allow issue parallel reset to the peripherals. This is self clearing bit.	W	0h	RESET _SYS

4.9.4 SYSTEM SLEEP CONTROL REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	SLEEP_IMMEDIATE 0 = System will only allow entry into sleep after PLL locks. 1 = System will allow entry into Heavy Sleep before PLL locks. Heavy Sleep : Any sleep state where the PLL is OFF. Light Sleep : Any sleep state where the PLL is ON.	R/W	0h	RESET_SYS
7:4	Reserved	RES	-	-
3	SLEEP_ALL By setting this bit to '1b' and then issuing a WAIT FOR INTERRUPT instruction, the EC can initiate the System Sleep mode. When no device requires the main system clock, the system enters the sleep mode defined by the field SLEEP_MODE. This bit is automatically cleared when the processor vectors to an interrupt. 1=Assert all sleep enables 0=Do not sleep all	R/W	0h	RESET_SYS
2	TEST Test bit. Should always be written with a '0b'.	R/W	0h	RESET_SYS
1	Reserved	RES	-	-
0	SLEEP_MODE Sleep modes differ only in the time it takes for the 48MHz clock domain to lock to 48MHz. The wake latency in all sleep modes is 0ms. Table 4-9 shows the time to lock latency for the different sleep modes. 1=Heavy Sleep 0=Light Sleep	R/W	0h	RESET_SYS

TABLE 4-9: SYSTEM SLEEP MODES

SLEEP_MODE	Sleep State	Latency to Lock	Description
0	LIGHT SLEEP	0	Output of the PLL is gated in sleep. The PLL remains on.
1	HEAVY SLEEP	3ms	The PLL is shut down while in sleep.

4.9.5 PROCESSOR CLOCK CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	PROCESSOR_CLOCK_DIVIDE The following list shows examples of settings for this field and the resulting EC clock rate. 48=divide the 48MHz clock by 48 (1MHz processor clock) 16=divide the 48MHz clock by 16 (4MHz processor clock) 4=divide the 48MHz clock by 4 (12MHz processor clock) 3=divide the 48MHz clock by 3 (16MHz processor clock) 1=divide the 48MHz clock by 1 (48MHz processor clock) No other values are supported.	R/W	4h	RESET _SYS

4.9.6 SLOW CLOCK CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	RES	-	-
9:0	SLOW_CLOCK_DIVIDE Configures the 100KHz clock domain. n=Divide by n 0=Clock off The default setting is for 100KHz.	R/W	1E0h	RESET _SYS

4.9.7 OSCILLATOR ID REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	PLL_LOCK Phase Lock Loop Lock Status	R	0h	RESET _SYS
7:0	TEST	R	N/A	RESET _SYS

4.9.8 PCR POWER RESET STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:12	Reserved	RES	-	-
10	32K_ACTIVE 1=The 32K clock input is present. The internal 32K clock is derived from the pin and the ring oscillator is synchronized to the external 32K clock 0=The 32K clock input is not present. The internal 32K clock is derived from the ring oscillator	R	-	RESET_SYS
9	Reserved	RES	-	-
8	WDT_EVENT This bit allows the application code to determine WDT_EVENT against RESET_VTR	R/W1C	0h	RESET_SYS-_nWDT
7	JTAG_RST# Indicates the JTAG_RST# pin status. The JTAG TRST# input is gated off low when Boundary scan mode is enabled and will not be set in this mode.	R	-	RESET_SYS
6	RESET_SYS_STATUS Indicates the status of RESET_SYS. The bit will not clear if a write 1 is attempted at the same time that a RESET_VTR occurs; this way a reset event is never missed. 1=A reset occurred 0=No reset occurred since the last time this bit was cleared	R/WC	1h	RESET_SYS
5	VBAT_RESET_STATUS Indicates the status of RESET_VBAT. The bit will not clear if a write of '1'b is attempted at the same time that a VBAT_RST_N occurs, this way a reset event is never missed. 1=A reset occurred 0=No reset occurred while VTR_CORE was off or since the last time this bit was cleared	R/WC	-	RESET_SYS
4	RESET_VTR_STATUS Indicates the status of RESET_VTR event.	R/W1C	1h	RESET_VTR

Note 1: This read-only status bit always reflects the current status of the event and is not affected by any Reset events.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	RESET_HOST_STATUS Indicates the status of RESET_HOST. 1=Reset not active 0=Reset active	R	-	Note 1
1:0	Reserved	RES	-	-
Note 1:	This read-only status bit always reflects the current status of the event and is not affected by any Reset events.			

4.9.9 POWER RESET CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:	Reserved	RES	-	-
7:	Reserved	RES	-	-
0	PWR_INV This bit allows firmware to control when the Host receives an indication that the VCC power is valid, by controlling the state of the PWROK pin.	R / R/W	1h	RESET _SYS

4.9.10 SYSTEM RESET REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	SOFT_SYS_RESET A write of a ‘1’ to this bit will force an assertion of the RESET_SYS reset signal, resetting the device. A write of a ‘0’ has no effect. Reads always return ‘0’.	W	-	-
7:0	Reserved	RES	-	-

4.9.11 PERIPHERAL RESET LOCK REGISTER

Offset	84h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PCR_RST_EN _LOCK If the lock is enabled, the peripherals cannot be reset by writing to the Reset enable register. Once Unlocked the Registers remain in the unlocked state until FW re-locks it with the Lock pattern 0xA6382D4Dh = Lock Pattern 0xA6382D4Ch = Unlock Pattern	RW	A6382D4Dh	RESET_SYS

5.0 ARM M4 BASED EMBEDDED CONTROLLER

5.1 Introduction

This chapter contains a description of the ARM M4 Embedded Controller (EC).

The EC is built around an ARM® Cortex®-M4 Processor provided by Arm Ltd. (the “ARM M4 IP”). The ARM Cortex® M4 is a full-featured 32-bit embedded processor, implementing the ARMv7-M THUMB instruction set in hardware.

The ARM M4 IP is configured as a Von Neumann, Byte-Addressable, Little-Endian architecture. It provides a single unified 32-bit byte-level address, for a total direct addressing space of 4GByte. It has multiple bus interfaces, but these express priorities of access to the chip-level resources (Instruction Fetch vs. Data RAM vs. others), and they do not represent separate addressing spaces.

The ARM M4 is configured as follows.

- **Little-Endian** byte ordering is selected at all times
- **Bit Banding** is included for efficient bit-level access
- **Debug** features are included at “Ex+” level, defined as follows:
  - **DWT** Unit provides 4 Data Watchpoint comparators and Execution Monitoring
- **Trace** features are included at “Full” level, defined as follows:
  - **DWT** for reporting breakpoints and watchpoints
  - **ITM** for profiling and to timestamp and output messages from instrumented firmware builds
  - **ETM** for instruction tracing, and for enhanced reporting of Core and DWT events
  - The ARM-defined **HTM** trace feature is **not included**
- **NVIC** Interrupt controller with 8 priority levels and up to 240 individually-vectored interrupt inputs
  - A Microchip-defined Interrupt Aggregator function (at chip level) may be used to group multiple interrupts onto single NVIC inputs
  - The ARM-defined **WIC** feature is **not included**. The Microchip Interrupt Aggregator function (at chip level) provides Wake control
- Single entry **Write Buffer** is incorporated

5.2 References

1. ARM Limited: Cortex®-M4 Technical Reference Manual, DDI0439C, 29 June 2010
2. ARM Limited: ARM®v7-M Architecture Reference Manual, DDI0403D, November 2010
3. NOTE: Filename DDI0403D_arm_architecture_v7m_reference_manual_errata_markup_1_0.pdf
4. ARM® Generic Interrupt Controller Architecture version 1.0 Architecture Specification, IHI0048A, September 2008
5. ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999
6. ARM Limited: AMBA® 3 AHB-Lite Protocol Specification, IHI0033A, 6 June 2006
7. ARM Limited: AMBA® 3 ATB Protocol Specification, IHI0032A, 19 June 2006
8. ARM Limited: Cortex-M™ System Design Kit Technical Reference Manual, DDI0479B, 16 June 2011
9. ARM Limited: CoreSight™ v1.0 Architecture Specification, IHI0029B, 24 March 2005
10. ARM Limited: CoreSight™ Components Technical Reference Manual, DDI0314H, 10 July 2009
11. ARM Limited: ARM® Debug Interface v5 Architecture Specification, IHI0031A, 8 February 2006
12. ARM Limited: ARM® Debug Interface v5 Architecture Specification ADIv5.1 Supplement, DSA09-PRDC-008772, 17 August 2009
13. ARM Limited: Embedded Trace Macrocell™ (ETMv1.0 to ETMv3.5) Architecture Specification, IHI0014Q, 23 September 2011
14. ARM Limited: CoreSight™ ETM™-M4 Technical Reference Manual, DDI0440C, 29 June 2010

5.3 Terminology

5.3.1 ARM IP TERMS AND ACRONYMS

- AHB

  Advanced High-Performance Bus, a system-level on-chip **AMBA 2** bus standard. See Reference[5], ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999.

- AHB-AP

  AHB Access Port, the **AP** option selected by Microchip for the **DAP**

- AHB-Lite

  A Single-Master subset of the **AHB** bus standard: defined in the **AMBA 3** bus standard. See Reference[6], ARM Limited: AMBA® 3 AHB-Lite Protocol Specification, IHI0033A, 6 June 2006.

- AMBA

  The collective term for bus standards originated by ARM Limited.

  AMBA 3 defines the IP's **AHB-Lite** and **ATB** bus interfaces.

  AMBA 2 (AMBA Rev. 2.0) defines the EC's **AHB** bus interface.

- AP

  Any of the ports on the **DAP** subblock for accessing on-chip resources on behalf of the Debugger, independent of processor operations. A single **AHB-AP** option is currently selected for this function.

- APB

  Advanced Peripheral Bus, a limited 32-bit-only bus defined in **AMBA 2** for I/O register accesses. This term is relevant only to describe the **PPB** bus internal to the EC core. See Reference [5], ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999.

- ARMv7

  The identifying name for the general architecture implemented by the **Cortex-M** family of IP products.

  The **ARMv7** architecture has no relationship to the older “ARM 7” product line, which is classified as an “ARMv3” architecture, and is very different.

- ATB

  Interface standard for Trace data to the **TPIU** from **ETM** and/or **ITM** blocks, Defined in **AMBA 3**. See Reference[7], ARM Limited: AMBA® 3 ATB Protocol Specification, IHI0032A, 19 June 2006.

- Cortex-M4

  The ARM designation for the specific IP selected for this product: a Cortex M4 processor core

- DAP

  Debug Access Port, a subblock consisting of **DP** and **AP** subblocks.

- DP

  Any of the ports in the **DAP** subblock for connection to an off-chip Debugger. A single **SWJ-DP** option is currently selected for this function, providing **JTAG** connectivity.

- DWT

  Data Watchdog and Trace subblock. This contains comparators and counters used for data watchpoints and Core activity tracing.

- ETM

  Embedded Trace Macrocell subblock. Provides enhancements for Trace output reporting, mostly from the **DWT** subblock. It adds enhanced instruction tracing, filtering, triggering and timestamping.

- FPB

  FLASH Patch Breakpoint subblock. Provides either Remapping (Address substitution) or Breakpointing (Exception or Halt) for a set of Instruction addresses and Data addresses. See Section 8.3 of Reference [1], ARM Limited: Cortex®-M4 Technical Reference Manual, DDI0439C, 29 June 2010.

- HTM

  AHB Trace Macrocell. This is an optional subblock that is **not included**.
- ITM

  Instrumentation Trace Macrocell subblock. Provides a HW Trace interface for “printf”-style reports from instrumented firmware builds, with timestamping also provided.
- MEM-AP

  A generic term for an **AP** that connects to a memory-mapped bus on-chip. For this product, this term is synonymous with the AHB Access Port, **AHB-AP.**
- NVIC

  Nested Vectored Interrupt Controller subblock. Accepts external interrupt inputs. See References [2], ARM Limited: ARM®v7-M Architecture Reference Manual, DDI0403D, November 2010 and [4], ARM® Generic Interrupt Controller Architecture version 1.0 Architecture Specification, IHI0048A, September 2008.
- PPB

  Private Peripheral Bus: A specific **APB** bus with local connectivity within the EC.
- ROM Table

  A ROM-based data structure in the Debug section that allows an external Debugger and/or a FW monitor to determine which of the Debug features are present.
- SWJ-DP

  Serial Wire / **JTAG** Debug Port, the **DP** option selected by Microchip for the **DAP**.
- TPA

  Trace Port Analyzer: any off-chip device that uses the TPIU output.
- TPIU

  Trace Port Interface Unit subblock. Multiplexes and buffers Trace reports from the ETM and ITM subblocks.
- WIC

  Wake-Up Interrupt Controller. This is an optional subblock that is **not included**.

5.3.2 MICROCHIP TERMS AND ACRONYMS

- Interrupt Aggregator

  This is a module that may be present at the chip level, which can combine multiple interrupt sources onto single interrupt inputs at the EC, causing them to share a vector.
- PMU

  Processor Memory Unit, this is a module that may be present at the chip level containing any memory resources that are closely-coupled to the CEC1712 EC. It manages accesses from both the EC processor and chip-level bus masters.

5.4 ARM M4 IP Interfaces

This section defines only the interfaces to the ARM IP itself. For the interfaces of the entire block, see Section 5.5, "Block External Interfaces".

The CEC1712 IP has the following major external interfaces, as shown in Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram":

- ICode AHB-Lite Interface
- DCode AHB-Lite Interface
- System AHB-Lite Interface
- Debug (JTAG) Interface
- Trace Port Interface
- Interrupt Interface

The EC operates on the model of a single 32-bit addressing space of byte addresses (4Gbytes, Von Neumann architecture) with Little-Endian byte ordering. On the basis of an internal decoder (part of the Bus Matrix shown in Figure 5-1), it routes Read/Write/Fetch accesses to one of three external interfaces, or in some cases internally (shown as the PPB interface).

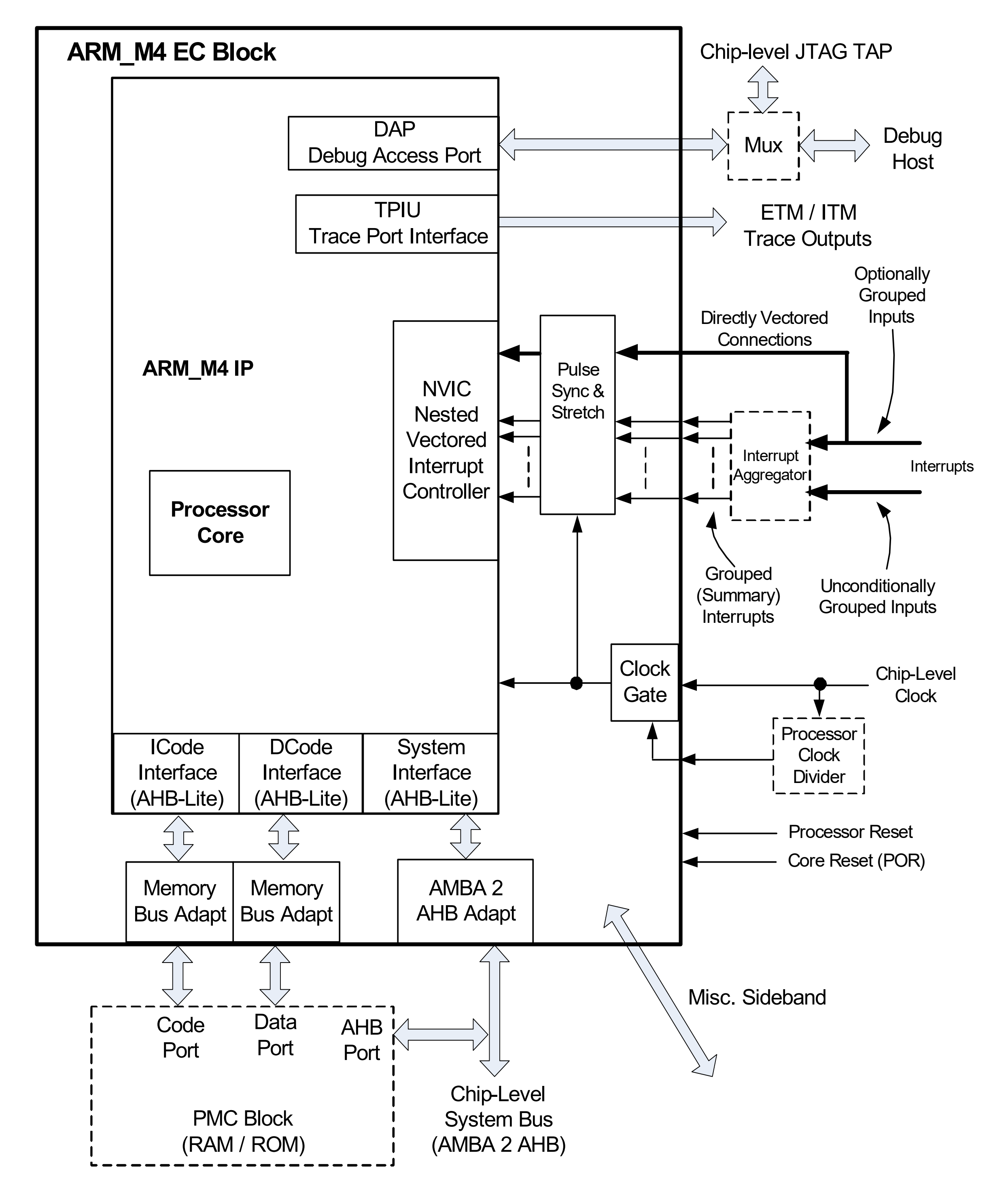
The EC executes instructions out of closely-coupled memory via the ICode Interface. Data accesses to closely-coupled memory are handled via the DCode Interface. The EC accesses the rest of the on-chip address space via the System AHB-Lite interface. The Debugger program in the host can probe the EC and all EC addressable memory via the JTAG debug interface.

Aliased addressing spaces are provided at the chip level so that specific bus interfaces can be selected explicitly where needed. For example, the EC's Bit Banding feature uses the System AHB-Lite bus to access resources normally accessed via the DCode or ICode interface.

Note: The EC executes most instructions in one clock cycle. If an instruction accesses code and data that are in different RAM blocks, then it takes one clock cycle to access both code and data (done in parallel). However, if the code and data blocks are in the same RAM block, then it takes two clock cycles (one clock for code access and one clock for data access) since it must do it sequentially.

5.5 Block External Interfaces

FIGURE 5-1: ARM M4 BASED EMBEDDED CONTROLLER I/O BLOCK DIAGRAM



5.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

5.6.1 POWER DOMAINS

TABLE 5-1: POWER SOURCES

Name	Description
VTR_CORE	The ARM M4 Based Embedded Controller is powered by VTR_CORE.

5.6.2 CLOCK INPUTS

5.6.2.1 Basic Clocking

The basic clocking comes from a free-running Clock signal provided from the chip level.

TABLE 5-2: CLOCK INPUTS

Name	Description
48MHz	The clock source to the EC. Division of the clock rate is determined by the PROCESSOR_CLOCK_DIVIDE field in the Processor Clock Control Register.

5.6.2.2 System Tick Clocking

The System Tick clocking is controlled by a signal from chip-level logic. It is the 48MHz divided by the following:

- ((PROCESSOR_CLOCK_DIVIDE)x2)+1

5.6.2.3 Debug JTAG Clocking

The Debug JTAG clocking comes from chip-level logic, which may multiplex or gate this clock. See Section 5.10, "Debugger Access Support".

5.6.2.4 Trace Clocking

The Clock for the Trace interface is identical to the 48MHz input.

5.6.3 RESETS

The reset interface from the chip level is given below.

TABLE 5-3: RESET SIGNALS

Name	Description
RESET_EC	The ARM M4 Based Embedded Controller is reset by RESET_EC.

5.7 Interrupts

The ARM M4 Based Embedded Controller is equipped with an Interrupt Interface to respond to interrupts. These inputs go to the IP's NVIC block after a small amount of hardware processing to ensure their detection at varying clock rates. See Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram".

As shown in Figure 5-1, an Interrupt Aggregator block may exist at the chip level, to allow multiple related interrupts to be grouped onto the same NVIC input, and so allowing them to be serviced using the same vector. This may allow the same interrupt handler to be invoked for a group of related interrupt inputs. It may also be used to expand the total number of interrupt inputs that can be serviced.

The NMI (Non-Maskable Interrupt) connection is tied off and not used.

5.7.1 NVIC INTERRUPT INTERFACE

The NVIC interrupt unit can be wired to up to 240 interrupt inputs from the chip level. The interrupts that are actually connected from the chip level are defined in the Interrupt section.

All NVIC interrupt inputs can be programmed as either pulse or level triggered. They can also be individually masked, and individually assigned to their own hardware-managed priority level.

5.7.2 NVIC RELATIONSHIP TO EXCEPTION VECTOR TABLE ENTRIES

The Vector Table consists of 4-byte entries, one per vector. Entry 0 is not a vector, but provides an initial Reset value for the Main Stack Pointer. Vectors start with the Reset vector, at Entry #1. Entries up through #15 are dedicated for internal exceptions, and do not involve the NVIC.

NVIC entries in the Vector Table start with Entry #16, so that NVIC Interrupt #0 is at Entry #16, and all NVIC interrupt numbers are incremented by 16 before accessing the Vector Table.

The number of connections to the NVIC determines the necessary minimum size of the Vector Table, as shown below. It can extend as far as 256 entries (255 vectors, plus the non-vector entry #0).

A Vector entry is used to load the Program Counter (PC) and the EPSR.T bit. Since the Program Counter only expresses code addresses in units of two-byte Halfwords, bit[0] of the vector location is used to load the EPSR.T bit instead, selecting THUMB mode for exception handling. Bit[0] must be ‘1’ in all vectors, otherwise a UsageFault exception will be posted (INVSTATE, unimplemented instruction set). If the Reset vector is at fault, the exception posted will be HardFault instead.

TABLE 5-4: EXCEPTION AND INTERRUPT VECTOR TABLE LAYOUT

Table Entry	Exception Number	Exception
Special Entry for Reset Stack Pointer		
0	(none)	Holds Reset Value for the Main Stack Pointer. Not a Vector.
Core Internal Exception Vectors start here		
1	1	Reset Vector (PC + EPSR.T bit)
2	2	NMI (Non-Maskable Interrupt) Vector
3	3	HardFault Vector
4	4	MemManage Vector
5	5	BusFault Vector
6	6	UsageFault Vector
7	(none)	(Reserved by ARM Ltd.)
8	(none)	(Reserved by ARM Ltd.)
9	(none)	(Reserved by ARM Ltd.)
10	(none)	(Reserved by ARM Ltd.)
11	11	SVCall Vector
12	12	Debug Monitor Vector
13	(none)	(Reserved by ARM Ltd.)
14	14	PendSV Vector
15	15	SysTick Vector
NVIC Interrupt Vectors start here		
16	16	NVIC Interrupt #0 Vector
. . .	. . .	. . .
n + 16	n + 16	NVIC Interrupt #n Vector
. . .	. . .	. . .
max + 16	max + 16	NVIC Interrupt #max Vector (Highest-numbered NVIC connection.)
. . .	. . .	. Table size may (but need not) extend further. . .
255	255	NVIC Interrupt #239 (Architectural Limit of Exception Table)

5.8 Low Power Modes

The ARM processor can enter Sleep or Deep Sleep modes internally. This action will cause an output signal Clock Required to be turned off, allowing clocks to be stopped from the chip level. However, Clock Required will still be held active, or set to active, unless all of the following conditions exist:

- No interrupt is pending.
- An input signal Sleep Enable from the chip level is active.
- The Debug JTAG port is inactive (reset or configured not present).

In addition, regardless of the above conditions, a chip-level input signal Force Halt may halt the processor and remove Clock Required.

5.9 Description

5.9.1 BUS CONNECTIONS

There are three bus connections used from CEC1712 EC block, which are directly related to the IP bus ports. See Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram".

For the mapping of addresses at the chip level, see Section 3.0, "Device Inventory".

5.9.1.1 Closely Coupled Instruction Fetch Bus

As shown in Figure 5-1, the AHB-Lite ICode port from the IP is converted to a more conventional SRAM memory-style bus and connected to the on-chip memory resources with routing priority appropriate to Instruction Fetches.

5.9.1.2 Closely Coupled Data Bus

As shown in Figure 5-1, the AHB-Lite DCode port from the IP is converted to a more conventional SRAM memory-style bus and connected to the on-chip memory resources with routing priority appropriate to fast Data Read/Write accesses.

5.9.1.3 Chip-Level System Bus

As shown in Figure 5-1, the AHB-Lite System port from the IP is converted from AHB-Lite to fully arbitrated multi-master capability (the AMBA 2 defined AHB bus: see Reference [5], ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999). Using this bus, all addressable on-chip resources are available. The multi-mastering capability supports the Microchip DMA and EMI features if present, as well as the Bit-Banding feature of the IP itself.

As also shown in Figure 5-1, the Closely-Coupled memory resources are also available through this bus connection using aliased addresses. This is required in order to allow Bit Banding to be used in these regions, but it also allows them to be accessed by DMA and other bus masters at the chip level.

Note:	Registers with properties such as Write-1-to-Clear (W1C), Read-to-Clear and FIFOs need to be handled with appropriate care when being used with the bit band alias addressing scheme. Accessing such a register through a bit band alias address will cause the hardware to perform a read-modify-write, and if a W1C-type bit is set, it will get cleared with such an access. For example, using a bit band access to the Interrupt Aggregator, including the Interrupt Enables and Block Interrupt Status to clear an IRQ will clear all active IRQs.

5.9.2 INSTRUCTION PIPELINING

There are no special considerations except as defined by ARM documentation.

5.10 Debugger Access Support

An external Debugger accesses the chip through a JTAG standard interface. The ARM Debug Access Port supports both the 2-pin SWD (Serial Wire Debug) interface and the 4-pin JTAG interface.

As shown in Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram", other resources at the chip level that share the JTAG port pins; for example chip-level Boundary Scan.

By default, debug access is disabled when the EC begins executing code. EC code enables debugging by writing the Debug Enable Register in the EC Subsystem Registers block.

TABLE 5-5: ARM JTAG ID

ARM Debug Mode	JTAG ID
SW-DP (2-wire)	0x2BA01477
JTAG (4-wire)	0x4BA00477

5.10.1 DEBUG AND ACCESS PORTS (SWJ-DP AND AHB-AP SUBBLOCKS)

These two subblocks work together to provide access to the chip for the Debugger using the Debug JTAG connection, as described in Chapter 4 of the ARM Limited: ARM® Debug Interface v5 Architecture Specification, IHI0031A, 8 February 2006.

5.10.2 BREAKPOINT, WATCHPOINT AND TRACE SUPPORT

See References [11], ARM Limited: ARM® Debug Interface v5 Architecture Specification, IHI0031A, 8 February 2006 and [12], ARM Limited: ARM® Debug Interface v5 Architecture Specification ADIv5.1 Supplement, DSA09-PRDC-008772, 17 August 2009. A summary of functionality follows.

Breakpoint and Watchpoint facilities can be programmed to do one of the following:

- Halt the processor. This means that the external Debugger will detect the event by periodically polling the state of the EC.
- Transfer control to an internal Debug Monitor firmware routine, by triggering the Debug Monitor exception (see Table 5-4, "Exception and Interrupt Vector Table Layout").

5.10.2.1 Instrumentation Support (ITM Subblock)

The Instrumentation Trace Macrocell (ITM) is for profiling software. This uses non-blocking register accesses, with a fixed low-intrusion overhead, and can be added to a Real-Time Operating System (RTOS), application, or exception handler. If necessary, product code can retain the register access instructions, avoiding probe effects.

5.10.2.2 HW Breakpoints and ROM Patching (FPB Subblock)

The Flash Patch and Breakpoint (FPB) block. This block can remap sections of ROM, typically Flash memory, to regions of RAM, and can set breakpoints on code in ROM. This block can be used for debug, and to provide a code or data patch to an application that requires field updates to a product in ROM.

5.10.2.3 Data Watchpoints and Trace (DWT Subblock)

The Debug Watchpoint and Trace (DWT) block provides watchpoint support, program counter sampling for performance monitoring, and embedded trace trigger control.

5.10.2.4 Trace Interface (ETM and TPIU)

The Embedded Trace Macrocell (ETM) provides instruction tracing capability. For details of functionality and usage, see References [13], ARM Limited: Embedded Trace Macrocell™ (ETMv1.0 to ETMv3.5) Architecture Specification, IHI0014Q, 23 September 2011 and [14], ARM Limited: CoreSight™ ETM™-M4 Technical Reference Manual, DDI0440C, 29 June 2010.

The Trace Port Interface Unit (TPIU) provides the external interface for the ITM, DWT and ETM.

5.11 Delay Register

5.11.1 DELAY REGISTER

Offset	1000_0000h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	RES	-	-
4:0	DELAY Writing a value *n,* from 0h to 31h, to this register will cause the ARM processor to stall for (*n*+1) microseconds (that is, from 1µS to 32µS). Reads will return the last value read immediately. There is no delay.	R/W	0h	RESET_ SYS

6.0 RAM AND ROM

6.1 SRAM

The CEC1712 contains two blocks of SRAM. The two SRAM blocks in the CEC1712 total 256KB. Both SRAM blocks can be used for either program or data accesses. Performance is enhanced when program fetches and data accesses are to different SRAM blocks, but a program will operate correctly even if both program and data accesses are targeting the same block simultaneously.

- The first SRAM, which is optimized for code access, is 224KB
- The second SRAM, which is optimized for data access, is32KB

6.2 ROM

The CEC1712 contains a 64KB block of ROM, located at address 00000000h in the ARM address space. The ROM contains boot code that is executed after the de-assertion of RESET_SYS. The boot code loads an executable code image into SRAM. The ROM also includes a set of API functions that can be used for cryptographic functions, as well as loading SRAM with programs or data.

6.3 Additional Memory Regions

6.3.1 ALIAS RAM

The Alias RAM region, starting at address 20000000h, is an alias of the SRAM located at 118000h, and is the same size as that SRAM block. EC software can access memory in either the primary address or in the alias region; however, access is considerably slower to the alias region. The alias region exists in order to enable the ARM bit-band region located at address 20000000h.

6.3.2 RAM BIT-BAND REGION

The RAM bit-band region is an alias of the SRAM located at 118000h, except that each bit is aliased to bit 0 of a 32-bit doubleword in the bit-band region. The upper 31 bits in each doubleword of the bit-band region are always 0. The bit-band region is therefore 32 times the size of the SRAM region. It can be used for atomic updates of individual bits of the SRAM, and is a feature of the ARM architecture.

The bit-band region can only be accessed by the ARM processor. Accesses by any other bus master will cause a memory fault.

6.3.3 CRYPTOGRAPHIC RAM

The cryptographic RAM is used by the cryptographic API functions in the ROM

6.3.4 REGISTER BIT-BAND REGION

The Register bit-band region is an 32-to-1 alias of the device register space starting at address 40000000h and ending with the Host register space at 400FFFFF. Every bit in the register space is aliased to a byte in the Register bit-band region, and like the RAM bit-band region, can be used by EC software to read and write individual register bits. Only the EC Device Registers and the GPIO Registers can be accessed via the bit-band region.

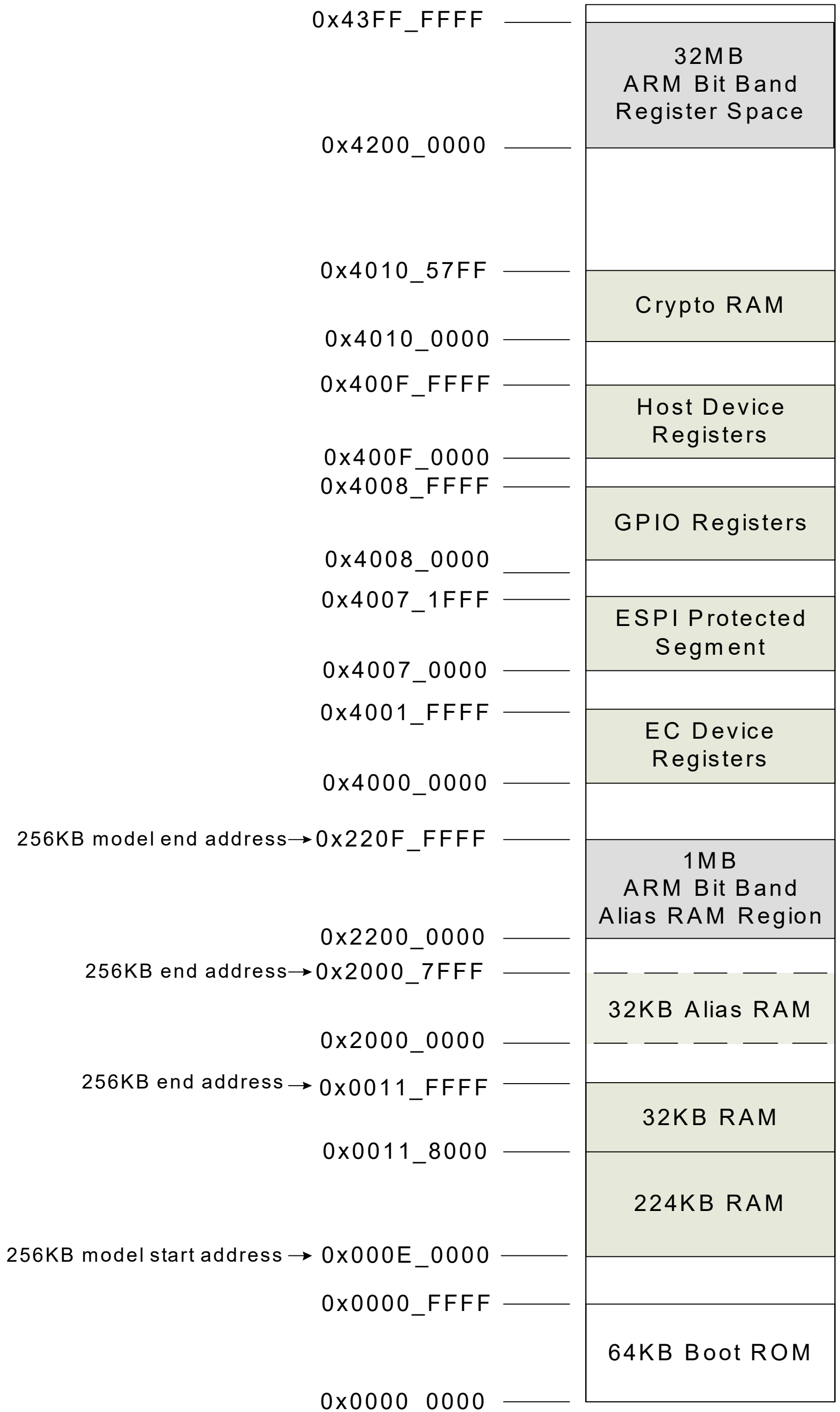
A one bit write operation to a register bit in the bit-band region is implemented by the ARM processor by performing a read, a bit modification, followed by a write back to the same register. Software must be careful when using bit-banding if a register contains bits have side effects triggered by a read.

The bit-band region can only be accessed by the ARM processor. Accesses by any other bus master will cause a memory fault.

6.4 Memory Map

The memory map of the RAM and ROM is represented as follows:

FIGURE 6-1: MEMORY LAYOUT



7.0 INTERNAL DMA CONTROLLER

7.1 Introduction

The Internal DMA Controller transfers data to/from the source from/to the destination. The firmware is responsible for setting up each channel. Afterwards either the firmware or the hardware may perform the flow control. The hardware flow control exists entirely inside the source device. Each transfer may be 1, 2, or 4 bytes in size, so long as the device supports a transfer of that size. Every device must be on the internal 32-bit address space.

7.2 References

No references have been cited for this chapter.

7.3 Terminology

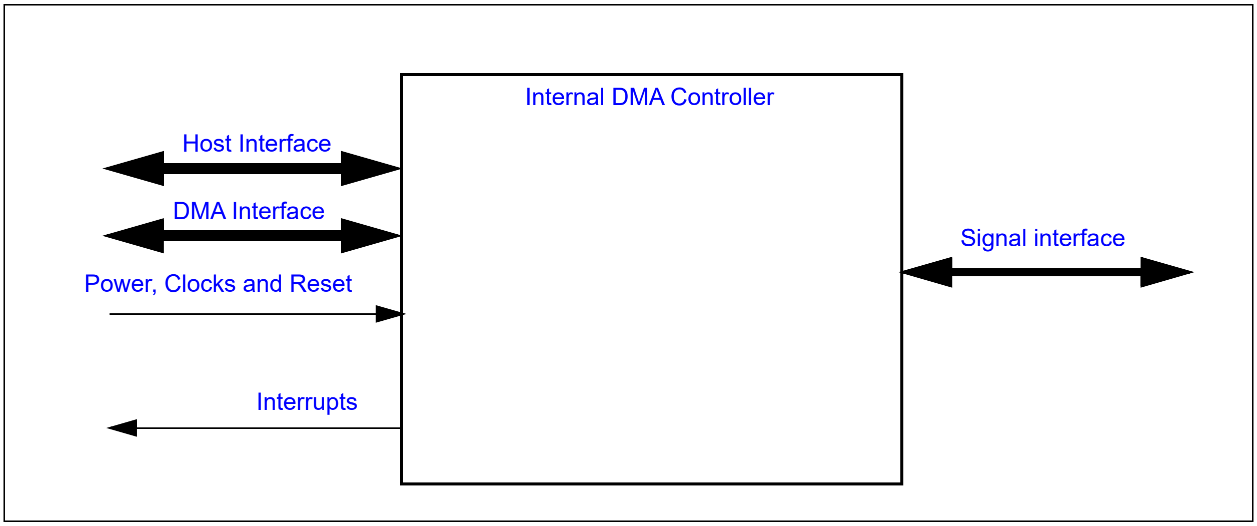
TABLE 7-1: TERMINOLOGY

Term	Definition
DMA Transfer	This is a complete **DMA Transfer** which is done after the **Master Device** terminates the transfer, the Firmware Aborts the transfer or the DMA reaches its transfer limit. A DMA Transfer may consist of one or more data packets.
Data Packet	Each data packet may be composed of 1, 2, or 4 bytes. The size of the data packet is limited by the max size supported by both the source and the destination. Both source and destination will transfer the same number of bytes per packet.
Channel	The Channel is responsible for end-to-end (source-to-destination) Data Packet delivery.
Device	A Device may refer to a Master or Slave connected to the DMA Channel. Each DMA Channel may be assigned one or more devices.
Master Device	This is the master of the DMA, which determines when it is active. The Firmware is the master while operating in Firmware Flow Control. The Hardware is the master while operating in Hardware Flow Control. The Master Device in Hardware Mode is selected by **DMA Channel Control:Hardware Flow Control Device**. It is the index of the **Flow Control** Port.
Slave Device	The Slave Device is defined as the device associated with the targeted Memory Address.
Source	The DMA Controller moves data from the Source to the Destination. The Source provides the data. The Source may be either the Master or Slave Controller.
Destination	The DMA Controller moves data from the Source to the Destination. The Destination receives the data. The Destination may be either the Master or Slave Controller.

7.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 7-1: INTERNAL DMA CONTROLLER I/O DIAGRAM



7.5 Signal interface

This block doesn't have any external signals that may be routed to the pin interface. This DMA Controller is intended to be used internally to transfer large amounts of data without the embedded controller being actively involved in the transfer.

7.6 Host Interface

The registers defined for the Internal DMA Controller are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

7.7 DMA Interface

Each DMA Master Device that may engage in a DMA transfer must have a compliant DMA interface. The following table lists the DMA Devices in the CEC1712.

TABLE 7-2: DMA CONTROLLER DEVICE SELECTION

Device Name	Device Number (Note 1)	Controller Source
SMB-I2C 0 Controller	0	Slave
	1	Master
SMB-I2C 1 Controller	2	Slave
	3	Master
SMB-I2C 2 Controller	4	Slave
	5	Master
SMB-I2C 3 Controller	6	Slave
	7	Master
SMB-I2C 4Controller	8	Transmit
	9	Receive

Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.

TABLE 7-2: DMA CONTROLLER DEVICE SELECTION (CONTINUED)

Device Name	Device Number (Note 1)	Controller Source
QMSPI Controller	10	Transmit
	11	Receive

Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.

TABLE 7-3: DMA CONTROLLER MASTER DEVICES SIGNAL LIST

Device Name	Dev Num	Device Signal Name	Direction	Description
SMB-I2C 0 Controller	0	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	1	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
SMB-I2C 1 Controller	2	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	3	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
SMB-I2C 2 Controller	4	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	5	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Master channel.

TABLE 7-3: DMA CONTROLLER MASTER DEVICES SIGNAL LIST (CONTINUED)

Device Name	Dev Num	Device Signal Name	Direction	Description
SMB-I2C 3 Controller	6	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	7	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
SMB-I2C 4 Controller	8	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	9	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
Quad SPI Controller	10	QSPI_TDMA_Req	INPUT	DMA request control from Quad SPI TX channel.
		QSPI_TDMA_Term	INPUT	DMA termination control from Quad SPI TX channel.
		QMSPI_TDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Quad SPI TDMA Channel.
	11	QSPI_RDMA_Req	INPUT	DMA request control from Quad SPI RX channel.
		QSPI_RDMA_Term	INPUT	DMA termination control from Quad SPI RX channel.
		QMSPI_RDMA_-Done	OUTPUT	DMA termination control from DMA Controller to Quad SPI RDMA Channel.

7.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

7.8.1 POWER DOMAINS

TABLE 7-4: POWER SOURCES

Name	Description
VTR_CORE	This power well sources the registers and logic in this block.

7.8.2 CLOCK INPUTS

TABLE 7-5: CLOCK INPUTS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

7.8.3 RESETS

TABLE 7-6: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.
RESET	This reset is generated if either the RESET_SYS is asserted or the SOFT_RESET bit is asserted.

7.9 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 7-7: INTERRUPTS

Source	Description
DMAx	Direct Memory Access Channel x This signal is generated by the STATUS_DONE bit.

7.10 Low Power Modes

The Internal DMA Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

When the block is commanded to go to sleep it will place the DMA block into sleep mode only after all transactions on the DMA have been completed. For Firmware Flow Controlled transactions, the DMA will wait until it hits its terminal count and clears the Go control bit. For Hardware Flow Control, the DMA will go to sleep after either the terminal count is hit, or the Master device flags the terminate signal.

7.11 Description

The CEC1712 features a 12 channel DMA controller. The DMA controller can autonomously move data from/to any DMA capable master device to/from any populated memory location. This mechanism allows hardware IP blocks to transfer large amounts of data into or out of memory without EC intervention.

The DMA has the following characteristics:

- Data is only moved 1 Data Packet at a time
- Data only moves between devices that are accessible via the internal 32-bit address space
- The DMA Controller has 12 DMA Channels
- Each DMA Channel may be configured to communicate with any DMA capable device on the 32-bit internal address space. Each device has been assigned a device number. See Section 7.7, "DMA Interface".

The controller will access SRAM buffers only with incrementing addresses (that is, it cannot start at the top of a buffer, nor does it handle circular buffers automatically). The controller does not handle chaining (that is, automatically starting a new DMA transfer when one finishes).

7.11.1 CONFIGURATION

The DMA Controller is enabled via the ACTIVATE bit in DMA Main Control Register register.

Each DMA Channel must also be individually enabled via the CHANNEL_ACTIVATE bit in the DMA Channel N Activate Register to be operational.

Before starting a DMA transaction on a DMA Channel the host must assign a DMA Master to the channel via HARDWARE_FLOW_CONTROL_DEVICE. The host must not configure two different channels to the same DMA Master at the same time.

Data will be transfered between the DMA Master, starting at the programmed DEVICE_ADDRESS, and the targeted memory location, starting at the MEMORY_START_ADDRESS. The address for either the DMA Master or the targeted memory location may remain static or it may increment. To enable the DMA Master to increment its address set the INCREMENT_DEVICE_ADDRESS bit. To enable the targeted memory location to increment its addresses set the INCREMENT_MEMORY_ADDRESS. The DMA transfer will continue as long as the target memory address being accessed is less than the MEMORY_END_ADDRESS. If the DMA Controller detects that the memory location it is attempting to access on the Target is equal to the MEMORY_END_ADDRESS it will notify the DMA Master that the transaction is done. Otherwise the Data will be transferred in packets. The size of the packet is determined by the TRANSFER_SIZE.

7.11.2 OPERATION

The DMA Controller is designed to move data from one memory location to another.

7.11.2.1 Establishing a Connection

A DMA Master will initiate a DMA Transaction by requesting access to a channel. The DMA arbiter, which evaluates each channel request using a basic round robin algorithm, will grant access to the DMA master. Once granted, the channel will hold the grant until it decides to release it, by notifying the DMA Controller that it is done.

If Firmware wants to prevent any other channels from being granted while it is active it can set the LOCK_CHANNEL bit.

7.11.2.2 Initiating a Transfer

Once a connection is established the DMA Master will issue a DMA request to start a DMA transfer. If Firmware wants to have a transfer request serviced it must set the RUN bit to have its transfer requests serviced.

Firmware can initiate a transaction by setting the TRANSFER_GO bit. The DMA transfer will remain active until either the Master issues a Terminate or the DMA Controller signals that the transfer is DONE. Firmware may terminate a transaction by setting the TRANSFER_ABORT bit.

Note:	Before initiating a DMA transaction via firmware the hardware flow control must be disabled via the DISABLE_HARDWARE_FLOW_CONTROL bit.

Data may be moved from the DMA Master to the targeted Memory address or from the targeted Memory Address to the DMA Master. The direction of the transfer is determined by the TRANSFER_DIRECTION bit.

Once a transaction has been initiated firmware can use the STATUS_DONE bit to determine when the transaction is completed. This status bit is routed to the interrupt interface. In the same register there are additional status bits that indicate if the transaction completed successfully or with errors. These bits are OR'd together with the STATUS_DONE bit to generate the interrupt event. Each status be may be individually enabled/disabled from generating this event.

7.11.2.3 Reusing a DMA Channel

After a DMA Channel controller has completed, firmware **must** clear both the DMA Channel N Control Register and the DMA Channel N Interrupt Status Register. After both have been cleared to 0, the Channel Control Register can then be configured for the next transaction.

7.11.2.4 CRC Generation

A CRC generator can be attached to a DMA channel in order to generate a CRC on the data as it is transfered from the source to the destination. The CRC used is the CRC-32 algorithm used in IEEE 802.3 and many other protocols, using the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. The CRC generation takes place in parallel with the data transfer; enabling CRC will not increase the time to complete a DMA transaction. The CRC generator has the optional ability to automatically transfer the generated CRC to the destination after the data transfer has completed.

CRC generation is subject to a number of restrictions:

- The CRC is only generated on channels that have the CRC hardware. See Table 7-10, "Channel Register Summary" for a definition of which channels have the ability to generate a CRC
- The DMA transfer must be 32-bits
- If CRC is enabled, DMA interrupts are inhibited until the CRC is completed, including the optional post-transfer copy of it is enabled
- The CRC must be initialized by firmware. The value FFFFFFFFh must be written to the Data Register in order to initialize the generator for the standard CRC-32-IEEE algorithm
- The CRC will be bit-order reversed and inverted as required by the CRC algorithm

7.11.2.5 Block Fill Option

A Fill engine can be attached to a DMA channel in order to provide a fast mechanism to set a block of memory to a fixed value (for example, clearing a block of memory to zero). The block fill operation runs approximately twice as fast as a memory-to-memory copy.

In order to fill memory with a constant value, firmware **must** configure the channel in the following order:

1. Set the DMA Channel N Fill Data Register to the desired fill value
2. Set the DMA Channel N Fill Enable Register to ‘1b’, enabling the Fill engine
3. Set the DMA Channel N Control Register to the following values:
   - RUN = 0
   - TRANSFER_DIRECTION = 0 (memory destination)
   - INCREMENT_MEMORY_ADDRESS = 1 (increment memory address after each transfer)
   - INCREMENT_DEVICE_ADDRESS = 1
   - DISABLE_HARDWARE_FLOW_CONTROL = 1 (no hardware flow control)
   - TRANSFER_SIZE = 1, 2 or 4 (as required)
   - TRANSFER_ABORT = 0
   - TRANSFER_GO = 1 (this starts the transfer)

7.12 EC Registers

The DMA Controller consists of a Main Block and a number of Channels. Table 7-9, "Main Register Summary" lists the registers in the Main Block and Table 7-10, "Channel Register Summary" lists the registers in each channel. Addresses for each register are determined by adding the offset to the Base Address for the DMA Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Registers are listed separately for the Main Block of the DMA Controller and for a DMA Channel. Each Channel has the same set of registers. The absolute register address for registers in each channel are defined by adding the Base Address for the DMA Controller Block, the Offset for the Channel shown in Table 7-8, "DMA Channel Offsets" to the offsets listed in Table 7-9, "Main Register Summary" or Table 7-10, "Channel Register Summary".

TABLE 7-8: DMA CHANNEL OFFSETS

Instance Name	Channel Number	Offset
DMA Controller	Main Block	000h
DMA Controller	0	040h
DMA Controller	1	080h
DMA Controller	2	0C0h
DMA Controller	3	100h
DMA Controller	4	140h
DMA Controller	5	180h
DMA Controller	6	1C0h
DMA Controller	7	200h
DMA Controller	8	240h
DMA Controller	9	280h

TABLE 7-8: DMA CHANNEL OFFSETS (CONTINUED)

Instance Name	Channel Number	Offset
DMA Controller	10	2C0h
DMA Controller	11	300h

TABLE 7-9: MAIN REGISTER SUMMARY

Offset	Register Name
00h	DMA Main Control Register
04h	DMA Data Packet Register

7.12.1 DMA MAIN CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
1	SOFT_RESET Soft reset the entire module. This bit is self-clearing.	W	0b	-
0	ACTIVATE Enable the blocks operation. 1=Enable block. Each individual channel must be enabled separately. 0=Disable all channels.	R/WS	0b	RESET

7.12.2 DMA DATA PACKET REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DATA_PACKET Debug register that has the data that is stored in the Data Packet. This data is read data from the currently active transfer source.	R	0000h	-

TABLE 7-10: CHANNEL REGISTER SUMMARY

Offset	Register Name (Note 1)
00h	DMA Channel N Activate Register
04h	DMA Channel N Memory Start Address Register
08h	DMA Channel N Memory End Address Register
0Ch	DMA Channel N Device Address
10h	DMA Channel N Control Register
14h	DMA Channel N Interrupt Status Register

Note 1: The letter ‘N’ following DMA Channel indicates the Channel Number. Each Channel implemented will have these registers to determine that channel’s operation.

2: These registers are only present on DMA Channel 0. They are reserved on all other channels.

3: These registers are only present on DMA Channel 1. They are reserved on all other channels.

TABLE 7-10: CHANNEL REGISTER SUMMARY (CONTINUED)

Offset	Register Name (Note 1)
18h	DMA Channel N Interrupt Enable Register
1Ch	TEST
20h (Note 2)	DMA Channel N CRC Enable Register
24h (Note 2)	DMA Channel N CRC Data Register
28h (Note 2)	DMA Channel N CRC Post Status Register
2Ch (Note 2)	TEST
20h (Note 3)	DMA Channel N Fill Enable Register
24h (Note 3)	DMA Channel N Fill Data Register
28h (Note 3)	DMA Channel N Fill Status Register
2Ch (Note 3)	TEST

Note 1: The letter ‘N’ following DMA Channel indicates the Channel Number. Each Channel implemented will have these registers to determine that channel’s operation.

2: These registers are only present on DMA Channel 0. They are reserved on all other channels.

3: These registers are only present on DMA Channel 1. They are reserved on all other channels.

7.12.3 DMA CHANNEL N ACTIVATE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	CHANNEL_ACTIVATE Enable this channel for operation. The DMA Main Control:Activate must also be enabled for this channel to be operational.	R/W	0h	RESET

7.12.4 DMA CHANNEL N MEMORY START ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MEMORY_START_ADDRESS This is the starting address for the Memory device. This field is updated by Hardware after every packet transfer by the size of the transfer, as defined by DMA Channel Control:Channel Transfer Size while the DMA Channel Control:Increment Memory Address is Enabled. The Memory device is defined as the device that is the slave device in the transfer. With Hardware Flow Control, the Memory device is the device that is not connected to the Hardware Flow Controlling device.	R/W	0000h	RESET

7.12.5 DMA CHANNEL N MEMORY END ADDRESS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MEMORY_END_ADDRESS This is the ending address for the Memory device. This will define the limit of the transfer, so long as DMA Channel Control:Increment Memory Address is Enabled. When the Memory Start Address is equal to this value, the DMA will terminate the transfer and flag the status DMA Channel Interrupt:Status Done. **Note:** If the TRANSFER_SIZE field in the DMA Channel N Control Register is set to 2 (for 2-byte transfers, this address **must** be evenly divisible by 2 or the transfer will not terminate properly. If the TRANSFER_SIZE field is set to 4 (for 4-byte transfers, this address **must** be evenly divisible by 4 or the transfer will not terminate properly.	R/W	0000h	RESET

7.12.6 DMA CHANNEL N DEVICE ADDRESS

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DEVICE_ADDRESS This is the Master Device address. This is used as the address that will access the Device on the DMA. The Device is defined as the Master of the DMA transfer; as in the device that is controlling the Hardware Flow Control. This field is updated by Hardware after every Data Packet transfer by the size of the transfer, as defined by DMA Channel Control:Transfer Size while the DMA Channel Control:Increment Device Address is Enabled.	R/W	0000h	RESET

7.12.7 DMA CHANNEL N CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	RES	-	-
25	TRANSFER_ABORT This is used to abort the current transfer on this DMA Channel. The aborted transfer will be forced to terminate immediately.	R/W	0h	RESET
24	TRANSFER_GO This is used for the **Firmware Flow Control** DMA transfer. This is used to start a transfer under the **Firmware Flow Control**. Do not use this in conjunction with the **Hardware Flow Control**; DISABLE_HARDWARE_FLOW_CONTROL must be set in order for this field to function correctly.	R/W	0h	RESET
23	Reserved	RES	-	-
22:20	TRANSFER_SIZE This is the transfer size in Bytes of each Data Packet transfer. The transfer size must be a legal transfer size. Valid sizes are 1, 2 and 4 Bytes.	R/W	0h	RESET
19	DISABLE_HARDWARE_FLOW_CONTROL Setting this bit to ‘1’b will Disable **Hardware Flow Control**. When disabled, any DMA Master device attempting to communicate to the DMA over the DMA Flow Control Interface will be ignored. This should be set before using the DMA channel in **Firmware Flow Control** mode.	R/W	0h	RESET
18	LOCK_CHANNEL This is used to lock the arbitration of the Channel Arbiter on this channel once this channel is granted. Once this is locked, it will remain on the arbiter until it has completed it transfer (either the Transfer Aborted, Transfer Done or Transfer Terminated conditions). **Note:** This setting may starve other channels if the locked channel takes an excessive period of time to complete.	R/W	0h	RESET
17	INCREMENT_DEVICE_ADDRESS If this bit is ‘1’b, the DEVICE_ADDRESS will be incremented by TRANSFER_SIZE after every Data Packet transfer	R/W	0h	RESET
16	INCREMENT_MEMORY_ADDRESS If this bit is ‘1’b, the MEMORY_START_ADDRESS will be incremented by TRANSFER_SIZE after every Data Packet transfer **Note:** If this is not set, the DMA will never terminate the transfer on its own. It will have to be terminated through the Hardware Flow Control or through a DMA Channel Control:Transfer Abort.	R/W	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:9	HARDWARE_FLOW_CONTROL_DEVICE This is the device that is connected to this channel as its Hardware Flow Control master. The Flow Control Interface is a bus with each master concatenated onto it. This selects which bus index of the concatenated Flow Control Interface bus is targeted towards this channel.	R/W	0h	RESET
8	TRANSFER_DIRECTION This determines the direction of the DMA Transfer. 1=Data Packet Read from MEMORY_START_ADDRESS followed by Data Packet Write to DEVICE_ADDRESS 0=Data Packet Read from DEVICE_ADDRESS followed by Data Packet Write to MEMORY_START_ADDRESS	R/W	0h	RESET
7:6	Reserved	RES	-	-
5	BUSY This is a status signal. 1=The DMA Channel is busy (FSM is not IDLE) 0=The DMA Channel is not busy (FSM is IDLE)	R	0h	RESET
4:3	TEST	R	0h	RESET
2	DONE This is a status signal. It is only valid while RUN is Enabled. This is the inverse of the **DMA Channel Control:Busy** field, except this is qualified with the **DMA Channel Control:Run** field. 1**=**Channel is done 0**=**Channel is not done or it is OFF	R	0h	RESET
1	REQUEST This is a status field. 1=There is a transfer request from the Master Device 0=There is no transfer request from the Master Device	R	0h	RESET
0	RUN This is a control field. It only applies to **Hardware Flow Control** mode. 1=This channel is enabled and will service transfer requests 0=This channel is disabled. All transfer requests are ignored	R/W	0h	RESET

7.12.8 DMA CHANNEL N INTERRUPT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	STATUS_DONE This is an interrupt source register. This flags when the DMA Channel has completed a transfer successfully on its side. A completed transfer is defined as when the DMA Channel reaches its limit; **Memory Start Address** equals **Memory End Address**. A completion due to a **Hardware Flow Control Terminate** will not flag this interrupt. 1=MEMORY_START_ADDRESS equals MEMORY_END_ADDRESS 0=MEMORY_START_ADDRESS does not equal MEMORY_END_ADDRESS	R/WC	0h	RESET
1	STATUS_ENABLE_FLOW_CONTROL This is an interrupt source register. This flags when the DMA Channel has encountered a **Hardware Flow Control Request** after the DMA Channel has completed the transfer. This means the Master Device is attempting to overflow the DMA. 1=Hardware Flow Control is requesting after the transfer has completed 0=No Hardware Flow Control event	R/WC	0h	RESET
0	STATUS_BUS_ERROR This is an interrupt source register. This flags when there is an Error detected over the internal 32-bit Bus. 1=Error detected.	R/WC	0h	RESET

7.12.9 DMA CHANNEL N INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	STATUS_ENABLE_DONE This is an interrupt enable for STATUS_DONE**.** 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET
1	STATUS_ENABLE_FLOW_CONTROL_ERROR This is an interrupt enable for STATUS_ENABLE_FLOW_CONTROL**.** 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	STATUS_ENABLE_BUS_ERROR This is an interrupt enable for STATUS_BUS_ERROR. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET

7.12.10 DMA CHANNEL N CRC ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	CRC_POST_TRANSFER_ENABLE The bit enables the transfer of the calculated CRC-32 after the completion of the DMA transaction. If the DMA transaction is aborted by either firmware or an internal bus error, the transfer will not occur. If the target of the DMA transfer is a device and the device signaled the termination of the DMA transaction, the CRC post transfer will not occur. 1=Enable the transfer of CRC-32 for DMA Channel N after the DMA transaction completes 0=Disable the automatic transfer of the CRC	R/W	0h	RESET
0	CRC_MODE_ENABLE 1=Enable the calculation of CRC-32 for DMA Channel N 0=Disable the calculation of CRC-32 for DMA Channel N	R/W	0h	RESET

7.12.11 DMA CHANNEL N CRC DATA REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CRC Writes to this register initialize the CRC generator. Reads from this register return the output of the CRC that is calculated from the data transfered by DMA Channel N. The output of the CRC generator is bit-reversed and inverted on reads, as required by the CRC-32-IEEE definition. A CRC can be accumulated across multiple DMA transactions on Channel N. If it is necessary to save the intermediate CRC value, the result of the read of this register must be bit-reversed and inverted before being written back to this register.	R/W	0h	RESET

7.12.12 DMA CHANNEL N CRC POST STATUS REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	CRC_DATA_READY This bit is set to ‘1b’ when the DMA controller is processing the post-transfer of the CRC data. This bit is cleared to ‘0b’ when the post-transfer completes.	R	0h	RESET
2	CRC_DATA_DONE This bit is set to ‘1b’ when the DMA controller has completed the post-transfer of the CRC data. This bit is cleared to ‘0b’ when the a new DMA transfer starts.	R	0h	RESET
1	CRC_RUNNING This bit is set to ‘1b’ when the DMA controller starts the post-transfer transmission of the CRC. It is only set when the post-transfer is enabled by the CRC_POST_TRANSFER_ENABLE field. This bit is cleared to ‘0b’ when the post-transfer completes.	R	0h	RESET
0	CRC_DONE This bit is set to ‘1b’ when the CRC calculation has completed from either normal or forced termination. It is cleared to ‘0b’ when the DMA controller starts a new transfer on the channel.	R	0h	RESET

7.12.13 DMA CHANNEL N FILL ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	FILL_MODE_ENABLE 1=Enable the Fill Engine for DMA Channel N 0=Disable the Fill Engine for DMA Channel N	R/W	0h	RESET

7.12.14 DMA CHANNEL N FILL DATA REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DATA This is the data pattern used to fill memory.	R/W	0h	RESET

7.12.15 DMA CHANNEL N FILL STATUS REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	FILL_RUNNING This bit is ‘1b’ when the Fill operation starts and is cleared to ‘0b’ when the Fill operation completes.	R	0h	RESET
0	FILL_DONE This bit is set to ‘1b’ when the Fill operation has completed from either normal or forced termination. It is cleared to ‘0b’ when the DMA controller starts a new transfer on the channel.	R	0h	RESET

8.0 EC INTERRUPT AGGREGATOR

8.1 Introduction

The EC Interrupt Aggregator works in conjunction with the processor's interrupt interface to handle hardware interrupts and exceptions.

Exceptions are synchronous to instructions, are not maskable, and have higher priority than interrupts. All three exceptions - reset, memory error, and instruction error - are hardwired directly to the processor. Interrupts are typically asynchronous and are maskable.

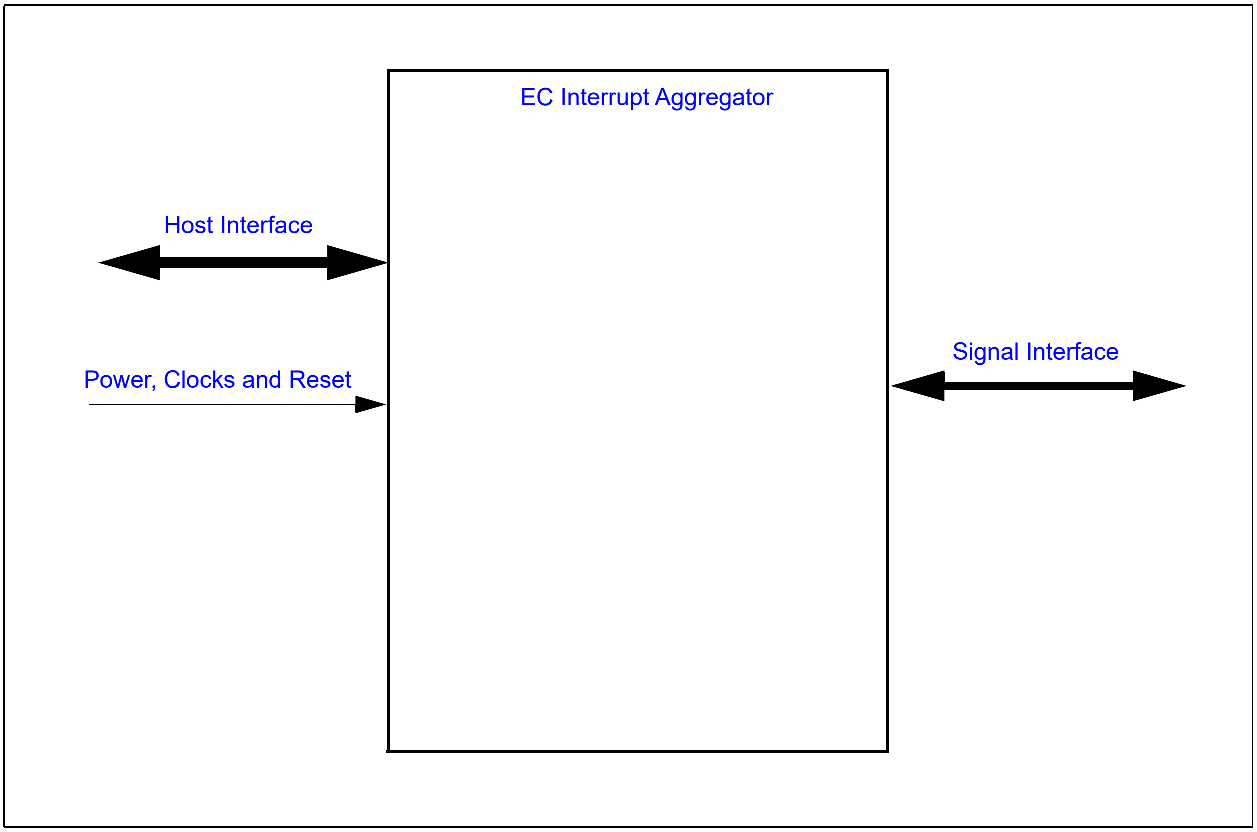
Interrupts classified as wake events can be recognized without a running clock, e.g., while the CEC1712 is in sleep state.

This chapter focuses on the EC Interrupt Aggregator. Please refer to embedded controller's documentation for more information on interrupt and exception handling.

8.2 Interface

This block is designed to be accessed internally via a registered host interface. The following diagram illustrates the various interfaces to the block.

FIGURE 8-1: EC INTERRUPT AGGREGATOR INTERFACE DIAGRAM



8.3 Signal Description

8.3.1 SIGNAL INTERFACE

There are no external signals for this block.

8.4 Host Interface

The registers defined for the EC Interrupt Aggregator are only accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

8.5 Power, Clocks and Reset

8.5.1 BLOCK POWER DOMAIN

TABLE 8-1: BLOCK POWER

Power Well Source	Effect on Block
VTR_CORE	The EC Interrupt Aggregator block and registers operate on this single power well.

8.5.2 BLOCK CLOCKS

TABLE 8-2: CLOCK INPUTS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

8.5.3 BLOCK RESET

TABLE 8-3: BLOCK RESETS

Reset Name	Reset Description
RESET_SYS	This signal is used to indicate when the VTR_CORE logic and registers in this block are reset.

8.6 Interrupts

This block aggregates all the interrupts targeted for the embedded controller into the Source Registers defined in Section 8.9, "EC Registers". The unmasked bits of each source register are then OR'd together and routed to the embedded controller's interrupt interface. The name of each Source Register identifies the IRQ number of the interrupt port on the embedded controller.

8.7 Low Power Modes

This block always automatically adjusts to operate in the lowest power mode by gating its clock when not required.

8.8 Description

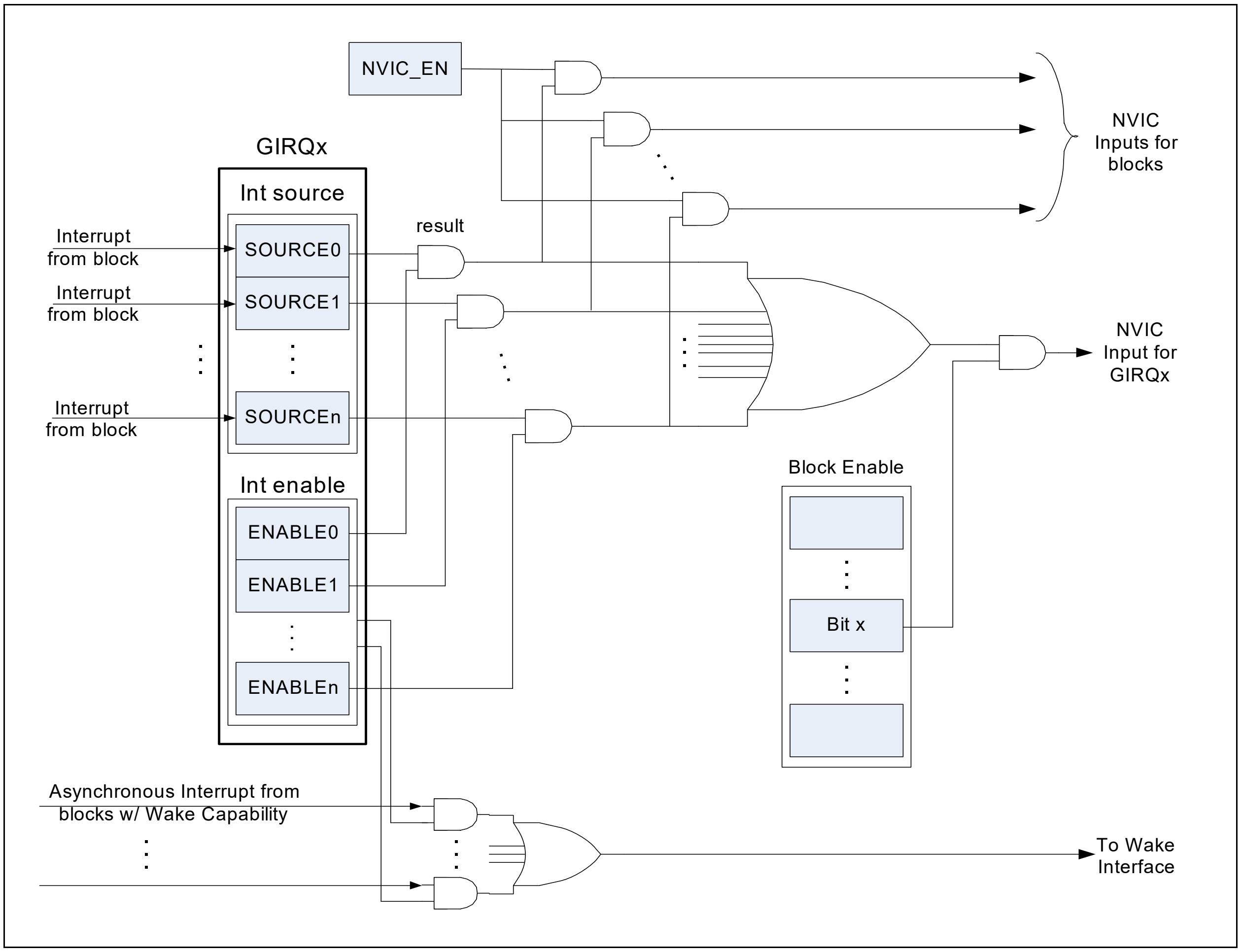
The interrupt generation logic is made of groups of signals, each of which consist of a Status register, a Enable Set register, and Enable Clear register and a Result register.

The Status and Enable are latched registers. There is one set of Enable register bits; both the Enable Set and Enable Clear registers return the same result when read. The Enable Set interface is used to set individual bits in the Enable register, and the Enable Clear is used to clear individual bits. The Result register is a bit by bit AND function of the Source and Enable registers. All the bits of the Result register are OR'ed together and AND'ed with the corresponding bit in the Block Select register to form the interrupt signal that is routed to the ARM interrupt controller.

The Result register bits may also be enabled to the NVIC block via the NVIC_EN bit in the Interrupt Control Register register. See Chapter 29.0, "EC Subsystem Registers"

Section 8.8.1 shows a representation of the interrupt structure.

FIGURE 8-2: INTERRUPT STRUCTURE



8.8.1 AGGREGATED INTERRUPTS

All interrupts are routed to the ARM processor through the ARM Nested Vectored Interrupt Controller (NVIC). As shown in Figure 8-2, "Interrupt Structure", all interrupt sources are aggregated into the GIRQx Source registers. In many cases, the Result bit for an individual interrupt source is tied directly to the NVIC. These interrupts are shown in the “Direct NVIC” column in the Interrupt Bit Assignments table. In addition, all GIRQx can also generate an interrupt to the NVIC when any of the enabled interrupts in its group is asserted. The NVIC vectors for the aggregated GIRQ interrupts are shown tin the “Agg NVIC” column.

Firmware should not enable the group GIRQ NVIC interrupt at the same time individual direct interrupts for members of the group are enabled. If both are enabled, the processor will receive two interrupts for an event, one from the GIRQ and one from the direct interrupt.

Note: The four Soft Interrupts that are defined by the RTOS Timer do not have individual NVIC vectors. If the use of the SWI interrupts is required, then all interrupts in the GIRQ must disable the individual NVIC vectors.

Note: These four Soft Interrupts are only available in aggregate mode

8.8.2 WAKE GENERATION

Wake-capable interrupts are listed in Table 3-3, "GIRQ_mapping" with a designation of 'Yes' in the Wake Event column. All interrupts, except GIRQ22, generate an EC Interrupt event. They are routed to source bits that are synchronized to the 32 MHz Ring Oscillator. If enabled, the Interrupt Result is fed into the Priority Encoder/Decision Logic, which generates the interrupt vector to the NVIC Interrupt Interface.

Some Interrupts, which are labeled Wake-Capable, are also routed as Wake Events to the Chip's Wake Logic. These are asynchronous events that are used to resume the 32 MHz Ring Oscillator operation from a sleep state and wake the processor.

8.8.2.1 Wake Capable Interrupts

All GPIO inputs are wake-capable. In order for a GPIO input to wake the CEC1712 from a sleep state, the Interrupt Detection field of the GPIO Pin Control Register must be set to Rising Edge Triggered, Falling Edge Triggered, or Either Edge Triggered. If the Interrupt Detection field is set to any other value, a GPIO input will not trigger a wake interrupt.

Some of the Wake Capable Interrupts are triggered by activity on pins that are shared with a GPIO. These interrupts will only trigger a wake if the Interrupt Detection field of the corresponding GPIO Pin Control Register is set to Rising Edge Triggered, Falling Edge Triggered, or Either Edge Triggered.

8.8.2.2 Wake-Only Events

Some devices which respond to an external master require the 48MHz clock domain to operate but do not necessarily require and immediate processing by the EC. Wake-only events provide the means to start the 48MHz clock domain without triggering an EC interrupt service routine. This events are grouped into a single GIRQ, GIRQ22. Events that are enabled in that GIRQ will start the clock domain when the event occurs, but will not invoke an EC interrupt. The SLEEP_ENABLE flags all remain asserted. If the activity for the event does not in turn trigger another EC interrupt, the CLOCK_REQUIRED for the block will re-assert and the configured sleep state will be re-entered.

8.8.3 INTERRUPT SUMMARY

Interrupt bit assignments, including wake capabilities and NVIC vector locations, are shown in the Interrupt Aggregator Bit Assignments Table in Section 3.0, "Device Inventory". The table lists all possible interrupt sources; the register bits for any interrupt source, such as a GPIO, that is not implemented in a particular part are reserved.

8.8.4 DISABLING INTERRUPTS

The Block Enable Clear Register and Block Enable Set Register should not be used for disabling and enabling interrupts for software operations i.e., critical sections. The ARM enable disable mechanisms should be used.

8.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for of the EC Interrupt Aggregator Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 8-4: REGISTER SUMMARY

Offset	Register Name
00h	GIRQ8 Source Register
04h	GIRQ8 Enable Set Register
08h	GIRQ8 Result Register
0Ch	GIRQ8 Enable Clear Register
10h	Reserved
14h	GIRQ9 Source Register
18h	GIRQ9 Enable Set Register
1Ch	GIRQ9 Result Register
20h	GIRQ9 Enable Clear Register
24h	Reserved

TABLE 8-4: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
28h	GIRQ10 Source Register
2Ch	GIRQ10 Enable Set Register
30h	GIRQ10 Result Register
34h	GIRQ10 Enable Clear Register
38h	Reserved
3Ch	GIRQ11 Source Register
40h	GIRQ11 Enable Set Register
44h	GIRQ11 Result Register
48h	GIRQ11 Enable Clear Register
4Ch	Reserved
50h	GIRQ12 Source Register
54h	GIRQ12 Enable Set Register
58h	GIRQ12 Result Register
5Ch	GIRQ12 Enable Clear Register
60h	Reserved
64h	GIRQ13 Source Register
68h	GIRQ13 Enable Set Register
6Ch	GIRQ13 Result Register
70h	GIRQ13 Enable Clear Register
74h	Reserved
78h	GIRQ14 Source Register
7Ch	GIRQ14 Enable Set Register
80h	GIRQ14 Result Register
84h	GIRQ14 Enable Clear Register
88h	Reserved
8Ch	GIRQ15 Source Register
90h	GIRQ15 Enable Set Register
94h	GIRQ15 Result Register
98h	GIRQ15 Enable Clear Register
9Ch	Reserved
A0h	GIRQ16 Source Register
A4h	GIRQ16 Enable Set Register
A8h	GIRQ16 Result Register
ACh	GIRQ16 Enable Clear Register
B0h	Reserved
B4h	GIRQ17 Source Register
B8h	GIRQ17 Enable Set Register
BCh	GIRQ17 Result Register
C0h	GIRQ17 Enable Clear Register
C4h	Reserved
C8h	GIRQ18 Source Register
CCh	GIRQ18 Enable Set Register
D0h	GIRQ18 Result Register
D4h	GIRQ18 Enable Clear Register

TABLE 8-4: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
D8h	Reserved
DCh	GIRQ19 Source Register
E0h	GIRQ19 Enable Set Register
E4h	GIRQ19 Result Register
E8h	GIRQ19 Enable Clear Register
ECh	Reserved
F0h	GIRQ20 Source Register
F4h	GIRQ20 Enable Set Register
F8h	GIRQ20 Result Register
FCh	GIRQ20 Enable Clear Register
100h	Reserved
104h	GIRQ21 Source Register
108h	GIRQ21 Enable Set Register
10Ch	GIRQ21 Result Register
110h	GIRQ21 Enable Clear Register
114h	Reserved
118h	GIRQ22 Source Register
11Ch	GIRQ22 Enable Set Register
120h	GIRQ22 Result Register
124h	GIRQ22 Enable Clear Register
128h	Reserved
12Ch	GIRQ23 Source Register
130h	GIRQ23 Enable Set Register
134h	GIRQ23 Result Register
138h	GIRQ23 Enable Clear Register
13Ch	Reserved
140h	GIRQ24 Source Register
144h	GIRQ24 Enable Set Register
148h	GIRQ24 Result Register
14Ch	GIRQ24 Enable Clear Register
150h	Reserved
154h	GIRQ25 Source Register
158h	GIRQ25 Enable Set Register
15Ch	GIRQ25 Result Register
160h	GIRQ25 Enable Clear Register
164h	Reserved
168h	GIRQ26 Source Register
16Ch	GIRQ26 Enable Set Register
170h	GIRQ26 Result Register
174h	GIRQ26 Enable Clear Register
200h	Block Enable Set Register
204h	Block Enable Clear Register
208h	Block IRQ Vector Register

All of the GIRQx Source, Enable Set, Enable Clear and Result registers have the same format. The following tables define the generic format for each of these registers. The bit definitions are defined in the sections that follow.

The behavior of the enable bit controlled by the GIRQx Enable Set and GIRQx Enable Clear Registers, the GIRQx Source bit, and the GIRQx Result bit is illustrated in Section 8.8.1, "Aggregated Interrupts".

8.9.1 GIRQ SOURCE REGISTERS

All of the GIRQx Source registers have the same format. The following table defines the generic format for each of these registers. The bit definitions are defined in the Interrupt Aggregator Bit Assignments Table in Section 3.0, "Device Inventory". Unassigned bits are Reserved and return 0.

> **Note:** If a GPIO listed in the tables does not appear in the pin list of a particular device, then the bits for that GPIO in the GIRQx Source, GIRQx Enable Clear, GIRQx Enable Set and GIRQx Result are reserved.

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	GIRQX_SOURCE The GIRQx Source bits are R/WC sticky status bits indicating the state of interrupt before the interrupt enable bit.	R/WC	0h	RESET_SYS

8.9.2 GIRQ ENABLE SET REGISTERS

All of the GIRQx Enable Set registers have the same format. The following table defines the generic format for each of these registers. Unassigned bits are Reserved and return 0.

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	GIRQX_ENABLE_SET Each GIRQx bit can be individually enabled to assert an interrupt event. Reads always return the current value of the internal GIRQX_ENABLE bit. The state of the GIRQX_ENABLE bit is determined by the corresponding GIRQX_ENABLE_SET bit and the GIRQX_ENABLE_CLEAR bit. (0=disabled, 1-enabled) 1=The corresponding interrupt in the GIRQx Source Register is enabled 0=No effect	R/WS	0h	RESET_SYS

8.9.3 GIRQ ENABLE CLEAR REGISTERS

All of the GIRQx Enable Clear registers have the same format. The following table defines the generic format for each of these registers. Unassigned bits are Reserved and return 0.

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	GIRQX_ENABLE_CLEAR Each GIRQx bit can be individually enabled to assert an interrupt event. Reads always return the current value of the internal GIRQX_ENABLE bit. The state of the GIRQX_ENABLE bit is determined by the corresponding GIRQX_ENABLE_SET bit and the GIRQX_ENABLE_CLEAR bit. (0=disabled, 1-enabled) 1=The corresponding interrupt in the GIRQx Source Register is disabled 0=No effect	R/WC	0h	RESET_SYS

8.9.4 GIRQ RESULT REGISTERS

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	1h	-
30:0	GIRQX_RESULT The GIRQX_RESULT bits are Read-Only status bits indicating the state of an interrupt. The RESULT is asserted ‘1’b when both the GIRQX_SOURCE bit and the corresponding GIRQX_ENABLE bit are ‘1’b.	R	0h	RESET_SYS

8.9.5 BLOCK ENABLE SET REGISTER

Offset	200h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:27	Reserved	RES	-	-

Offset	200h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
26:8	IRQ_VECTOR_ENABLE_SET Each bit in this field enables the group GIRQ interrupt assertion to the NVIC. 1=Interrupts in the GIRQx Source Register may be enabled 0=No effect	R/WS	0h	RESET_SYS
7:0	Reserved	RES	-	-

8.9.6 BLOCK ENABLE CLEAR REGISTER

Offset	204h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:27	Reserved	RES	-	-
26:8	IRQ_VECTOR_ENABLE_CLEAR Each bit in this field disables the group GIRQ interrupt assertion to the NVIC. 1=Interrupts in the GIRQx Source Register are disabled 0=No effect	R/WC	0h	RESET_SYS
7:0	Reserved	RES	-	-

8.9.7 BLOCK IRQ VECTOR REGISTER

Offset	208h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:27	Reserved	RES	0h	-
26:8	IRQ_VECTOR Each bit in this field reports the status of the group GIRQ interrupt assertion to the NVIC. If the GIRQx interrupt is disabled as a group, by the Block Enable Clear Register, then the corresponding bit will be ‘0’b and no interrupt will be asserted.	R	0h	RESET_SYS
7:0	Reserved	RES	0h	-

9.0 CHIP CONFIGURATION

9.1 Introduction

The Global Configuration Registers are use for chip-level configuration. The chip's Device ID and Revision are located in the Global Configuration space and may be used to uniquely identify this chip.

9.2 Configuration Registers

The EC address for each register is formed by adding the Base Address for Global Configuration block shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 9-1: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS

Register	Host Offset	Description
Chip (Global) Control Registers		
TEST	02h	TEST. This register location is reserved for Microchip use. Modifying this location may cause unwanted results.
Reserved	03h - 06h	Reserved - Writes are ignored, reads return 0.
Reserved	08h - 18h	Reserved - Writes are ignored, reads return 0.
Device Revision	1Ch	A read-only register which provides device revision information.
Device Sub ID	1Dh	Read-Only register which provides the device sub-identification.
Device ID[7:0]	1Eh	Read-Only register which provides Device ID LSB.
Device ID[15:8]	1Fh	Read-Only register which provides Device ID MSB.
Legacy Device ID	20h	A read-only register which provides Legacy device identification. The value of this register is FEh
TEST	22h - 23h	TEST. This register locations are reserved for Microchip use. Modifying these locations may cause unwanted results.
Reserved	24h	Reserved – writes are ignored, reads return "0".
TEST	25h - 2Fh	TEST. This register locations are reserved for Microchip use. Modifying these locations may cause unwanted results.

10.0 UART

10.1 Introduction

The 16550 UART (Universal Asynchronous Receiver/Transmitter) is a full-function Serial Port that supports the standard RS-232 Interface.

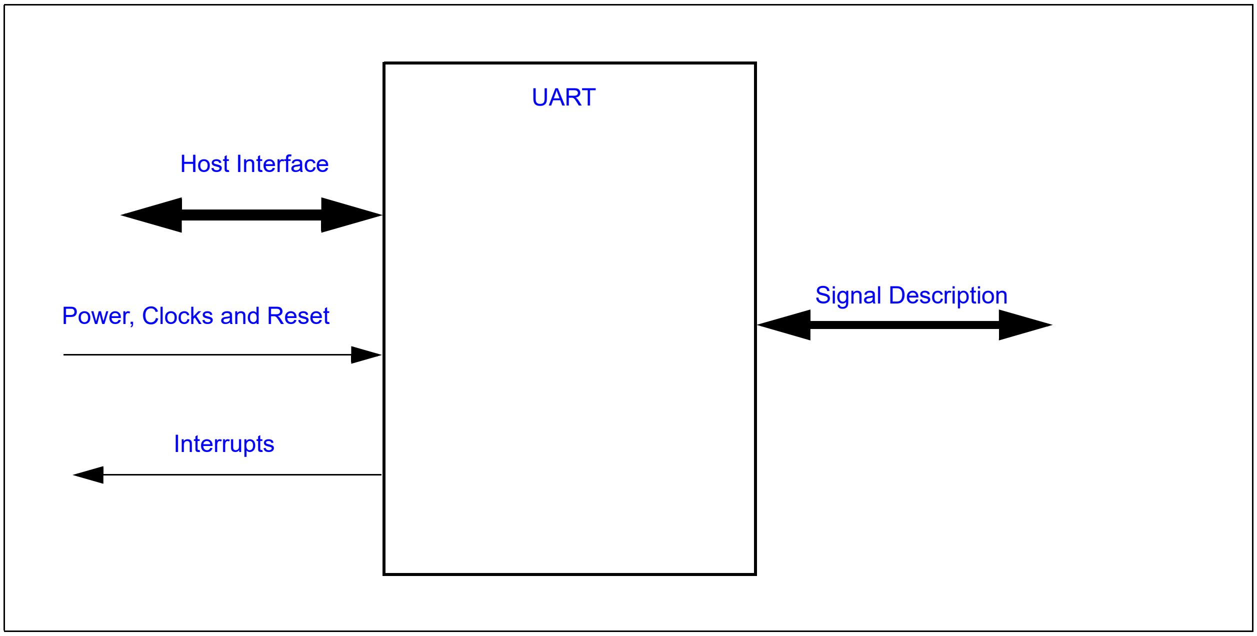
10.2 References

- EIA Standard RS-232-C specification

10.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 10-1: I/O DIAGRAM OF BLOCK



10.4 Signal Description

TABLE 10-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
DTR#	Output	Active low Data Terminal ready output for the Serial Port. Handshake output signal notifies modem that the UART is ready to transmit data. This signal can be programmed by writing to bit 1 of the Modem Control Register (MCR). **Note:** Defaults to tri-state on V3_DUAL power on.
DCD#	Input	Active low Data Carrier Detect input for the serial port. Handshake signal which notifies the UART that carrier signal is detected by the modem. The CPU can monitor the status of DCD# signal by reading bit 7 of Modem Status Register (MSR). A DCD# signal state change from low to high after the last MSR read will set MSR bit 3 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when DCD # changes state.

TABLE 10-1: SIGNAL DESCRIPTION TABLE (CONTINUED)

Name	Direction	Description
DSR#	Input	Active low Data Set Ready input for the serial port. Handshake signal which notifies the UART that the modem is ready to establish the communication link. The CPU can monitor the status of DSR# signal by reading bit 5 of Modem Status Register (MSR). A DSR# signal state change from low to high after the last MSR read will set MSR bit 1 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when DSR# changes state.
RI#	Input	Active low Ring Indicator input for the serial port. Handshake signal which notifies the UART that the telephone ring signal is detected by the modem. The CPU can monitor the status of RI# signal by reading bit 6 of Modem Status Register (MSR). A RI# signal state change from low to high after the last MSR read will set MSR bit 2 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when RI# changes state.
RTS#	Output	Active low Request to Send output for the Serial Port. Handshake output signal notifies modem that the UART is ready to transmit data. This signal can be programmed by writing to bit 1 of the Modem Control Register (MCR). The hardware reset will reset the RTS# signal to inactive mode (high). RTS# is forced inactive during loop mode operation. Defaults to tri-state on V3_DUAL power on.
CTS#	Input	Active low Clear to Send input for the serial port. Handshake signal which notifies the UART that the modem is ready to receive data. The CPU can monitor the status of CTS# signal by reading bit 4 of Modem Status Register (MSR). A CTS# signal state change from low to high after the last MSR read will set MSR bit 0 to a 1. If bit 3 of the Interrupt Enable Register is set, the interrupt is generated when CTS# changes state. The CTS# signal has no effect on the transmitter.
TXD	Output	Transmit serial data output.
RXD	Input	Receiver serial data input.

10.5 Host Interface

The registers defined for UART is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

10.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

10.6.1 POWER DOMAINS

TABLE 10-2: POWER SOURCES

Name	Description
VTR_CORE	This Power Well is used to power the registers and logic in this block.

10.6.2 CLOCKS

TABLE 10-3: CLOCK INPUTS

Name	Description
UART_CLK	An external clock that may be used as an alternative to the internally-generated 1.8432MHz and 48MHz baud clocks. Selection between internal baud clocks and an external baud clock is configured by the CLK_SRC bit in the Configuration Select Register.
48MHz	This is the main clock domain. Because the clock input must be within ± 2% in order to generate standard baud rates, the 48MHz clock must be generated by a reference clock with better than 1% accuracy and locked to its frequency before the UART will work with the standard rates.

TABLE 10-4: BAUD CLOCKS

Name	Description
1.8432MHz	The UART requires a 1.8432 MHz ± 2% clock input for baud rate generation of standard baud rates up to 115,200 baud. It is derived from the system 48MHz clock domain.
48MHz	It may be used as an alternative to the 1.8432MHz clock, generating non-standard baud rates up to 1,500,000 baud.

10.6.3 RESETS

TABLE 10-5: RESET SIGNALS

Name	Description
RESET_SYS	This reset is asserted when VTR_CORE is applied.
RESET_HOST	This is an alternate reset condition, typically asserted when the main power rail is asserted.
RESET	This reset is determined by the POWER bit signal. When the power bit signal is 1, this signal is equal to RESET_VCC, if present. When the power bit signal is 0, this signal is equal to RESET_SYS.

10.7 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 10-6: SYSTEM INTERRUPTS

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 10-12, "Interrupt Control Table".

TABLE 10-7: EC INTERRUPTS

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 10-12, "Interrupt Control Table".

10.8 Low Power Modes

The UART may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry.

10.9 Description

The UART is compatible with the 16450, the 16450 ACE registers and the 16C550A. The UART performs serial-to-parallel conversions on received characters and parallel-to-serial conversions on transmit characters. Two sets of baud rates are provided. When the 1.8432 MHz source clock is selected, standard baud rates from 50 to 115.2K are available. When the source clock is 48MHz, baud rates up to 1,500K are available. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. The UART contains a programmable baud rate generator that is capable of dividing the input clock signal by 1 to 32767. The UART is also capable of supporting the MIDI data rate. Refer to the Configuration Registers for information on disabling, powering down and changing the base address of the UART. The UART interrupt is enabled by programming OUT2 of the UART to logic “1.” Because OUT2 is logic “0,” it disables the UART's interrupt. The UART is accessible by both the Host and the EC.

10.9.1 PROGRAMMABLE BAUD RATE

The Serial Port contains a programmable Baud Rate Generator that is capable of dividing the internal clock source by any divisor from 1 to 32767. Unless an external clock source is configured, the clock source is either the 1.8432MHz clock source or the 48MHz clock source. The output frequency of the Baud Rate Generator is 16x the Baud rate. Two eight bit latches store the divisor in 16 bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16 bit Baud counter is immediately loaded. This prevents long counts on initial load. If a 0 is loaded into the BRG registers, the output divides the clock by the number 3. If a 1 is loaded, the output is the inverse of the input oscillator. If a two is loaded, the output is a divide by 2 signal with a 50% duty cycle. If a 3 or greater is loaded, the output is low for 2 bits and high for the remainder of the count.

The following tables show possible baud rates.

TABLE 10-8: UART BAUD RATES USING CLOCK SOURCE 1.8432MHz

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
50	0	2304
75	0	1536
110	0	1047
134.5	0	857
150	0	768
300	0	384
600	0	192
1200	0	96
1800	0	64
2000	0	58
2400	0	48
3600	0	32
4800	0	24
7200	0	16
9600	0	12
19200	0	6
38400	0	3
57600	0	2
115200	0	1

TABLE 10-9: UART BAUD RATES USING CLOCK SOURCE 48MHz

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
125000	1	24
136400	1	22
150000	1	20
166700	1	18
187500	1	16
214300	1	14
250000	1	12
300000	1	10
375000	1	8
500000	1	6
750000	1	4
1500000	1	2
3000000	1	1

10.10 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the UART. Host access for each register listed in this table is defined as an offset in the Host address space to the Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the UART shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 10-10: RUNTIME REGISTER SUMMARY

DLAB Note 1	Offset	Register Name
0	0h	Receive Buffer Register
0	0h	Transmit Buffer Register
1	0h	Programmable Baud Rate Generator LSB Register
1	1h	Programmable Baud Rate Generator MSB Register
0	1h	Interrupt Enable Register
x	02h	FIFO Control Register
x	02h	Interrupt Identification Register
x	03h	Line Control Register
x	04h	Modem Control Register
x	05h	Line Status Register
x	06h	Modem Status Register
x	07h	Scratchpad Register

Note 1: DLAB is bit 7 of the Line Control Register.

10.10.1 RECEIVE BUFFER REGISTER

Offset	0h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	RECEIVED_DATA This register holds the received incoming data byte. Bit 0 is the least significant bit, which is transmitted and received first. Received data is double buffered; this uses an additional shift register to receive the serial data stream and convert it to a parallel 8 bit word which is transferred to the Receive Buffer register. The shift register is not accessible.	R	0h	RESET

10.10.2 TRANSMIT BUFFER REGISTER

Offset	0h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	TRANSMIT_DATA This register contains the data byte to be transmitted. The transmit buffer is double buffered, utilizing an additional shift register (not accessible) to convert the 8 bit data word to a serial format. This shift register is loaded from the Transmit Buffer when the transmission of the previous byte is complete.	W	0h	RESET

10.10.3 PROGRAMMABLE BAUD RATE GENERATOR LSB REGISTER

Offset	00h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	BAUD_RATE_DIVISOR_LSB See Section 10.9.1, "Programmable Baud Rate".	R/W	0h	RESET

10.10.4 PROGRAMMABLE BAUD RATE GENERATOR MSB REGISTER

Offset	01h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	BAUD_CLK_SEL If CLK_SRC is ‘0’: • 0=The baud clock is derived from the 1.8432MHz. • 1=IThe baud clock is derived from the 48MHz. If CLK_SRC is ‘1’: • This bit has no effect	R/W	0h	RESET
6:0	BAUD_RATE_DIVISOR_MSB See Section 10.9.1, "Programmable Baud Rate".	R/W	0h	RESET

10.10.5 INTERRUPT ENABLE REGISTER

The lower four bits of this register control the enables of the five interrupt sources of the Serial Port interrupt. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of this register. Similarly, setting the appropriate bits of this register to a high, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and disables any Serial Port interrupt out of the CEC1712. All other system functions operate in their normal manner, including the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are described below.

Offset	01h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	Reserved	RES	-	-
3	EMSI This bit enables the MODEM Status Interrupt when set to logic “1”. This is caused when one of the Modem Status Register bits changes state.	R/W	0h	RESET
2	ELSI This bit enables the Received Line Status Interrupt when set to logic “1”. The error sources causing the interrupt are Overrun, Parity, Framing and Break. The Line Status Register must be read to determine the source.	R/W	0h	RESET
1	ETHREI This bit enables the Transmitter Holding Register Empty Interrupt when set to logic “1”.	R/W	0h	RESET
0	ERDAI This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic “1”.	R/W	0h	RESET

10.10.6 FIFO CONTROL REGISTER

This is a write only register at the same location as the Interrupt Identification Register.

Note: DMA is not supported.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	RECV_FIFO_TRIGGER_LEVEL These bits are used to set the trigger level for the RCVR FIFO interrupt.	W	0h	RESET
5:4	Reserved	RES	-	-
3	DMA_MODE_SELECT Writing to this bit has no effect on the operation of the UART. The RXRDY and TXRDY pins are not available on this chip.	W	0h	RESET
2	CLEAR_XMIT_FIFO Setting this bit to a logic “1” clears all bytes in the XMIT FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.	W	0h	RESET
1	CLEAR_RECv_FIFO Setting this bit to a logic “1” clears all bytes in the RCVR FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.	W	0h	RESET
0	EXRF Enable XMIT and RECV FIFO. Setting this bit to a logic “1” enables both the XMIT and RCVR FIFOs. Clearing this bit to a logic “0” disables both the XMIT and RCVR FIFOs and clears all bytes from both FIFOs. When changing from FIFO Mode to non-FIFO (16450) mode, data is automatically cleared from the FIFOs. This bit must be a 1 when other bits in this register are written to or they will not be properly programmed.	W	0h	RESET

TABLE 10-11: RECV FIFO TRIGGER LEVELS

Bit 7	Bit 6	RECV FIFO Trigger Level (BYTES)
0	0	1
	1	4
1	0	8
	1	14

10.10.7 INTERRUPT IDENTIFICATION REGISTER

By accessing this register, the host CPU can determine the highest priority interrupt and its source. Four levels of priority interrupt exist. They are in descending order of priority:

1. Receiver Line Status (highest priority)
2. Received Data Ready

3. Transmitter Holding Register Empty
4. MODEM Status (lowest priority)

Information indicating that a prioritized interrupt is pending and the source of that interrupt is stored in the Interrupt Identification Register (refer to Table 10-12). When the CPU accesses the IIR, the Serial Port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. During this CPU access, even if the Serial Port records new interrupts, the current indication does not change until access is completed. The contents of the IIR are described below.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	FIFO_EN These two bits are set when the FIFO CONTROL Register bit 0 equals 1.	R	0h	RESET
5:4	Reserved	RES	-	-
3:1	INTID These bits identify the highest priority interrupt pending as indicated by Table 10-12, "Interrupt Control Table". In non-FIFO mode, Bit[3] is a logic “0”. In FIFO mode Bit[3] is set along with Bit[2] when a timeout interrupt is pending.	R	0h	RESET
0	IPEND This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic ‘0’ an interrupt is pending and the contents of the IIR may be used as a pointer to the appropriate internal service routine. When bit 0 is a logic ‘1’ no interrupt is pending.	R	1h	RESET

TABLE 10-12: INTERRUPT CONTROL TABLE

FIFO Mode Only	Interrupt Identification Register			Interrupt SET and RESET Functions			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	0	1	-	None	None	-
	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the Line Status Register
		0		Second	Received Data Available	Receiver Data Available	Read Receiver Buffer or the FIFO drops below the trigger level.
1					Character Timeout Indication	No Characters Have Been Removed From or Input to the RCVR FIFO during the last 4 Char times and there is at least 1 char in it during this time	Reading the Receiver Buffer Register
0	0	1		Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if Source of Interrupt) or Writing the Transmitter Holding Register
	0	0		Fourth	MODEM Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect	Reading the MODEM Status Register

10.10.8 LINE CONTROL REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	DLAB Divisor Latch Access Bit (DLAB). It must be set high (logic “1”) to access the Divisor Latches of the Baud Rate Generator during read or write operations. It must be set low (logic “0”) to access the Receiver Buffer Register, the Transmitter Holding Register, or the Interrupt Enable Register.	R/W	0h	RESET
6	BREAK_CONTROL Set Break Control bit. When bit 6 is a logic “1”, the transmit data output (TXD) is forced to the Spacing or logic “0” state and remains there (until reset by a low level bit 6) regardless of other transmitter activity. This feature enables the Serial Port to alert a terminal in a communications system.	R/W	0h	RESET
5	STICK_PARITY Stick Parity bit. When parity is enabled it is used in conjunction with bit 4 to select Mark or Space Parity. When LCR bits 3, 4 and 5 are 1 the Parity bit is transmitted and checked as a 0 (Space Parity). If bits 3 and 5 are 1 and bit 4 is a 0, then the Parity bit is transmitted and checked as 1 (Mark Parity). If bit 5 is 0 Stick Parity is disabled. Bit 3 is a logic “1” and bit 5 is a logic “1”, the parity bit is transmitted and then detected by the receiver in the opposite state indicated by bit 4.	R/W	0h	RESET
4	PARITY_SELECT Even Parity Select bit. When bit 3 is a logic “1” and bit 4 is a logic “0”, an odd number of logic “1”'s is transmitted or checked in the data word bits and the parity bit. When bit 3 is a logic “1” and bit 4 is a logic “1” an even number of logic “1”s is transmitted and checked.	R/W	0h	RESET
3	ENABLE_PARITY Parity Enable bit. When bit 3 is a logic “1”, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and the first stop bit of the serial data. (The parity bit is used to generate an even or odd number of 1s when the data word bits and the parity bit are summed).	R/W	0h	RESET
2	STOP_BITS This bit specifies the number of stop bits in each transmitted or received serial character. Table 10-13 summarizes the information. The receiver will ignore all stop bits beyond the first, regardless of the number used in transmitting.	R/W	0h	RESET
1:0	WORD_LENGTH These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:	R/W	0h	RESET

TABLE 10-13: STOP BITS

Bit 2	Word Length	Number of Stop Bits
0	--	1
1	5 bits	1.5
	6 bits	2
	7 bits	
	8 bits	

TABLE 10-14: SERIAL CHARACTER

Bit 1	Bit 0	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

The Start, Stop and Parity bits are not included in the word length.

10.10.9 MODEM CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	Reserved	RES	-	-
4	LOOPBACK This bit provides the loopback feature for diagnostic testing of the Serial Port. When bit 4 is set to logic “1”, the following occur: 1. The TXD is set to the Marking State (logic “1”). 2. The receiver Serial Input (RXD) is disconnected. 3. The output of the Transmitter Shift Register is “looped back” into the Receiver Shift Register input. 4. All MODEM Control inputs (CTS#, DSR#, RI# and DCD#) are disconnected. 5. The four MODEM Control outputs (DTR#, RTS#, OUT1 and OUT2) are internally connected to the four MODEM Control inputs (DSR#, CTS#, RI#, DCD#). 6. The Modem Control output pins are forced inactive high. 7. Data that is transmitted is immediately received. This feature allows the processor to verify the transmit and receive data paths of the Serial Port. In the diagnostic mode, the receiver and the transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.	R/W	0h	RESET
3	OUT2 Output 2 (OUT2). This bit is used to enable an UART interrupt. When OUT2 is a logic “0”, the serial port interrupt output is forced to a high impedance state - disabled. When OUT2 is a logic “1”, the serial port interrupt outputs are enabled.	R/W	0h	RESET
2	OUT1 This bit controls the Output 1 (OUT1) bit. This bit does not have an output pin and can only be read or written by the CPU.	R/W	0h	RESET
1	RTS This bit controls the Request To Send (RTS#) output. When bit 1 is set to a logic “1”, the RTS# output is forced to a logic “0”. When bit 1 is set to a logic “0”, the RTS# output is forced to a logic “1”.	R/W	0h	RESET
0	DTR This bit controls the Data Terminal Ready (DTR#) output. When bit 0 is set to a logic “1”, the DTR# output is forced to a logic “0”. When bit 0 is a logic “0”, the DTR# output is forced to a logic “1”.	R/W	0h	RESET

10.10.10 LINE STATUS REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	FIFO_ERROR This bit is permanently set to logic “0” in the 450 mode. In the FIFO mode, this bit is set to a logic “1” when there is at least one parity error, framing error or break indication in the FIFO. This bit is cleared when the LSR is read if there are no subsequent errors in the FIFO.	R	0h	RESET
6	TRANSMIT_ERROR Transmitter Empty. Bit 6 is set to a logic “1” whenever the Transmitter Holding Register (THR) and Transmitter Shift Register (TSR) are both empty. It is reset to logic “0” whenever either the THR or TSR contains a data character. Bit 6 is a read only bit. In the FIFO mode this bit is set whenever the THR and TSR are both empty,	R	0h	RESET
5	TRANSMIT_EMPTY Transmitter Holding Register Empty Bit 5 indicates that the Serial Port is ready to accept a new character for transmission. In addition, this bit causes the Serial Port to issue an interrupt when the Transmitter Holding Register interrupt enable is set high. The THRE bit is set to a logic “1” when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic “0” whenever the CPU loads the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty, it is cleared when at least 1 byte is written to the XMIT FIFO. Bit 5 is a read only bit.	R	0h	RESET
4	BREAK_INTERRUPT Break Interrupt. Bit 4 is set to a logic “1” whenever the received data input is held in the Spacing state (logic “0”) for longer than a full word transmission time (that is, the total time of the start bit + data bits + parity bits + stop bits). The BI is reset after the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. Restarting after a break is received, requires the serial data (RXD) to be logic “1” for at least 1/2 bit time. Bits 1 through 4 are the error conditions that produce a Receiver Line Status Interrupt BIT 3 whenever any of the corresponding conditions are detected and the interrupt is enabled	R	0h	RESET

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	FRAME_ERROR Framing Error. Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to a logic “1” whenever the stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level). This bit is reset to a logic “0” whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. The Serial Port will try to resynchronize after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this 'start' bit twice and then takes in the 'data'.	R	0h	RESET
2	PARITY ERROR Parity Error. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. This bit is set to a logic “1” upon detection of a parity error and is reset to a logic “0” whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO.	R	0h	RESET
1	OVERRUN_ERROR Overrun Error. Bit 1 indicates that data in the Receiver Buffer Register was not read before the next character was transferred into the register, thereby destroying the previous character. In FIFO mode, an overrun error will occur only when the FIFO is full and the next character has been completely received in the shift register, the character in the shift register is overwritten but not transferred to the FIFO. This bit is set to a logic “1” immediately upon detection of an overrun condition, and reset whenever the Line Status Register is read.	R	0h	RESET
0	DATA_READY Data Ready. It is set to a logic ‘1’ whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to a logic ‘0’ by reading all of the data in the Receive Buffer Register or the FIFO.	R	0h	RESET

10.10.11 MODEM STATUS REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	DCD This bit is the complement of the Data Carrier Detect (DCD#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to OUT2 in the MCR.	R	0h	RESET
6	RI This bit is the complement of the Ring Indicator (RI#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to OUT1 in the MCR.	R	0h	RESET
5	DSR This bit is the complement of the Data Set Ready (DSR#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to DTR# in the MCR.	R	0h	RESET
4	CTS This bit is the complement of the Clear To Send (CTS#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to RTS# in the MCR.	R	0h	RESET
3	DDCD Delta Data Carrier Detect (DDCD). Bit 3 indicates that the DCD# input to the chip has changed state. NOTE: Whenever bit 0, 1, 2, or 3 is set to a logic ‘1’, a MODEM Status Interrupt is generated.	R	0h	RESET
2	TERI Trailing Edge of Ring Indicator (TERI). Bit 2 indicates that the RI# input has changed from logic ‘0’ to logic ‘1’.	R	0h	RESET
1	DDSR Delta Data Set Ready (DDSR). Bit 1 indicates that the DSR# input has changed state since the last time the MSR was read.	R	0h	RESET
0	DCTS Delta Clear To Send (DCTS). Bit 0 indicates that the CTS# input to the chip has changed state since the last time the MSR was read.	R	0h	RESET

10.10.12 SCRATCHPAD REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SCRATCH This 8 bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.	R/W	0h	RESET

10.11 Configuration Registers

Configuration Registers for an instance of the UART are listed in the following table. Host access to Configuration Registers is through the Configuration Port using the Logical Device Number of each instance of the UART and the Index shown in the “Host Index” column of the table. The EC can access Configuration Registers directly. The EC address for each register is formed by adding the Base Address for each instance of the UART shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “EC Offset” column.

TABLE 10-15: CONFIGURATION REGISTER SUMMARY

EC Offset	Host Index	Register Name
330h	30h	Activate Register
3F0h	F0h	Configuration Select Register

10.11.1 ACTIVATE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	ACTIVATE When this bit is 1, the UART logical device is powered and functional. When this bit is 0, the UART logical device is powered down and inactive.	R/W	0b	RESET

10.11.2 CONFIGURATION SELECT REGISTER

Offset	F0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	POLARITY 1=The UART_TX and UART_RX pins functions are inverted 0=The UART_TX and UART_RX pins functions are not inverted	R/W	0b	RESET
1	POWER 1=The RESET reset signal is derived from RESET_HOST 0=The RESET reset signal is derived from RESET_SYS	R/W	1b	RESET
0	CLK_SRC 1=The UART Baud Clock is derived from an external clock source 0=The UART Baud Clock is derived from one of the two internal clock sources	R/W	0b	RESET

11.0 GPIO INTERFACE

11.1 General Description

The CEC1712 GPIO Interface provides general purpose input monitoring and output control, as well as managing many aspects of pin functionality; including, multi-function Pin Multiplexing Control, GPIO Direction control, PU/PD (PU_PD) resistors, asynchronous wakeup and synchronous Interrupt Detection (int_det), GPIO Direction, and Polarity control, as well as control of pin drive strength and slew rate.

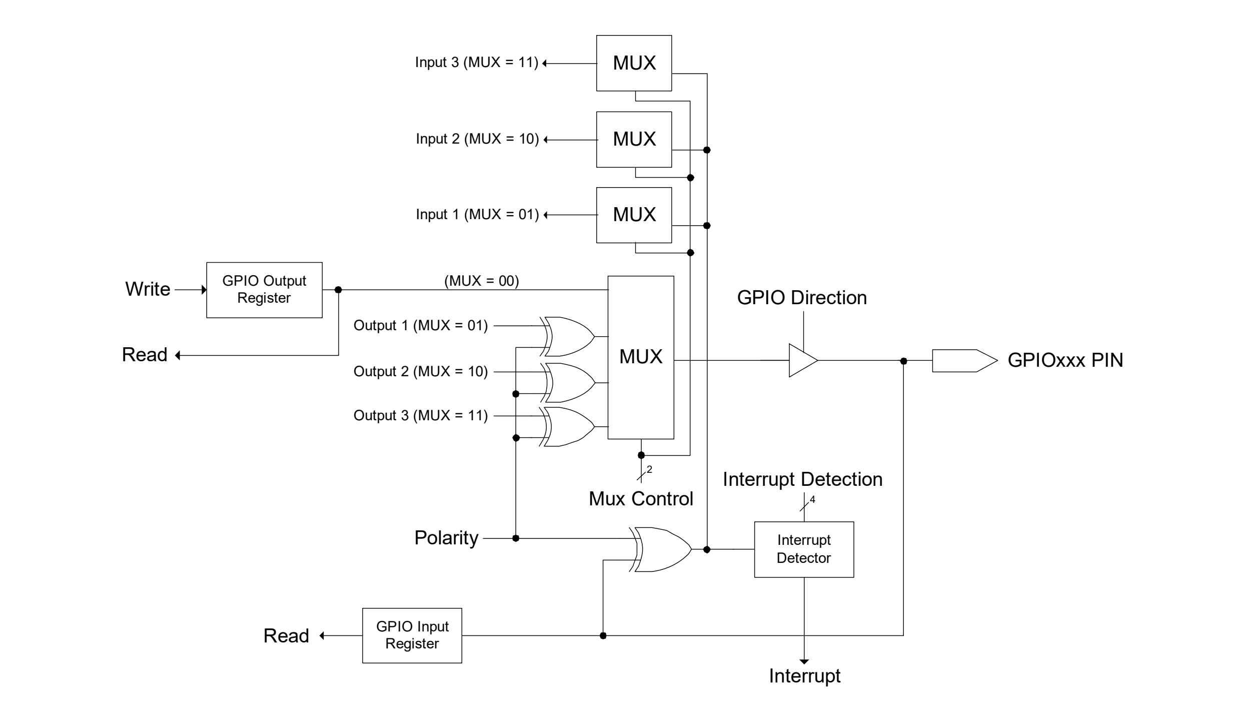
Features of the GPIO Interface include:

- Inputs:
  - Asynchronous rising and falling edge wakeup detection
  - Interrupt High or Low Level
- On Output:
  - Push Pull or Open Drain output
- Pull up or pull down resistor control
- Interrupt and wake capability available for all GPIOs
- Programmable pin drive strength and slew rate limiting
- Group- or individual control of GPIO data.
- Multiplexing of all multi-function pins are controlled by the GPIO interface

11.2 Block Diagram

The GPIO Interface Block Diagram shown in Figure 11-1 illustrates the functionality of a single CEC1712 GPIO Interface pin. The source for the Pin Multiplexing Control, Interrupt Detection (int_det), GPIO Direction, and Polarity controls in Figure 11-1 is a Pin Control Register that is associated with each pin (see Section 11.6.1.1, "Pin Control Register," on page 143).

FIGURE 11-1: GPIO INTERFACE BLOCK DIAGRAM



11.3 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

11.3.1 POWER DOMAINS

Name	Description
VTR_CORE	The registers and logic in this block are powered by VTR_CORE.

11.3.2 CLOCK INPUTS

Name	Description
48MHz	The 48MHz is used for synchronizing the GPIO inputs.

11.3.3 RESETS

Name	Description
RESET_SYS	This reset is asserted when VTR_CORE is applied.
RESET_VCC	This is an alternate reset condition, typically asserted when the main power rail is asserted.

11.4 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
GPIO_Event	Each pin in the GPIO Interface has the ability to generate an interrupt event. This event may be used as a wake event. The GPIO Interface can generate an interrupt source event on a high level, low level, rising edge and falling edge, as configured by the Interrupt Detection (int_det) bits in the Pin Control Register associated with the GPIO signal function. **Note:** The minimum pulse width required to generate an interrupt/wakeup event is 5ns.

11.5 Description

The GPIO Interface refers to all the GPIOxxx pins implemented in the design. GPIO stands for General Purpose I/O.

The GPIO signals may be used by firmware to both monitor and control a pin in “bit-banged” mode. The GPIOs may be individually controlled via their Pin Control Register or group controlled via the Output and Input GPIO registers. The GPIO Output Control Select

The GPIO Pin control registers are used to select the alternate functions on GPIO pins (unless otherwise specified), to control the buffer direction, strength, and polarity, to control the internal pull-ups and pull-downs, for VCC emulation, and for selecting the event type that causes a GPIO interrupt.

The GPIO input is always live, even when an alternate function is selected. Firmware may read the GPIO input anytime to see the value on the pin. In addition, the GPIO interrupt is always functional, and may be used for either the GPIO itself or to support the alternate functions on the pin. See FIGURE 11-1: GPIO Interface Block Diagram on page 138.

11.5.1 ACCESSING GPIOS

There are two ways to access GPIO output data. Bit [10] is used to determine which GPIO output data bit affects the GPIO output pin.

- Grouped Output GPIO Data
  - Outputs to individual GPIO ports are grouped into 32-bit GPIO Output Registers.
- Individual GPIO output data
  - Alternatively, each GPIO output port is individually accessible via Bit [16] in the port's Pin Control Register. On reads, Bit [16] returns the programmed value, not the value on the pin.

There are two ways to access GPIO input data.

- Input GPIO Data
  - Inputs from individual GPIO ports are grouped into 32-bit GPIO Input Registers and always reflect the current state of the GPIO input from the pad.
- GPIO input from pad
  - Alternatively, each GPIO input port is individually accessible via Bit [24] in the port's Pin Control Register. Bit [24] always reflects the current state of GPIO input from the pad.

11.5.2 GPIO INDEXING

Each GPIO signal function name consists of a 4-character prefix ("GPIO") followed by a 3-digit octal-encoded index number.   In the CEC1712 GPIO indexing is done sequentially starting from 'GPIO000.'

11.5.3 PIN CONTROL REGISTERS

Each GPIO has two Pin Control registers. The Pin Control Register, which is the primary register, is used to read the value of the input data and set the output either high or low. It is used to select the alternate function via the Mux Control bits, set the Polarity of the input, configure and enable the output buffer, configure the GPIO interrupt event source, enable internal pull-up/pull-down resistors, and to enable VCC Emulation via the Power Gating Signals (PGS) control bits. The Pin Control Register 2 is used to configure the output buffer drive strength and slew rate.

The following tables define the default settings for the two Pin Control registers for each GPIO in each product group.

11.5.3.1 Pin Control Register Defaults

Please refer to Section 3.5, "GPIO Register Assignments" for the Pin Control Register default information.

11.6 GPIO Registers

The registers listed in the Register Summary table are for a single instance of the CEC1712. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the Register Base Address Table.

TABLE 11-1: REGISTER BASE ADDRESS TABLE

Instance Name	Instance Number	Host	Address Space	Base Address
GPIO	0	EC	32-bit internal address space	4008_1000h Note 11-1

Note 11-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

Note 11-2 The GPIO registers may be accessed by the eSPI Host via the EMI block via GPIO commands or by direct access if enabled by firmware. See the firmware documentation for a description of this access method.

Note: Registers and bits associated with GPIOs not implemented are Reserved. Please refer to Section 2.3, "Pin List" for GPIOs implemented in the chip.

TABLE 11-2: REGISTER SUMMARY

Offset	Register Name
000h - 01Ch	GPIO000-GPIO007 Pin Control Register
020h - 03Ch	GPIO010-GPIO017 Pin Control Register
040h - 05Ch	GPIO020-GPIO027 Pin Control Register
060h - 078h	GPIO030-GPIO036 Pin Control Register
080h - 09Ch	GPIO040-GPIO047 Pin Control Register
0A0h - 0BCh	GPIO050-GPIO057 Pin Control Register
0C0h - 0DCh	GPIO060-GPIO067 Pin Control Register
0E0h - 0F8h	GPIO070-GPIO077 Pin Control Register
100h - 11Ch	GPIO100-GPIO107 Pin Control Register
128h - 13Ch	GPIO112-GPIO117 Pin Control Register
140h - 15Ch	GPIO120-GPIO127 Pin Control Register
160h - 16Ch	GPIO130-GPIO137 Pin Control Register
180h - 19Ch	GPIO140-GPIO147 Pin Control Register
1A0h - 1BCh	GPIO150-GPIO157 Pin Control Register
1C0h - 1DCh	GPIO160-GPIO167 Pin Control Register
1E0h - 1F4h	GPIO170-GPIO177 Pin Control Register
200h - 21Ch	GPIO200-GPIO207 Pin Control Register
220h - 23Ch	GPIO210-GPIO217 Pin Control Register
240h - 25Ch	GPIO221-GPIO227 Pin Control Register
260h - 27Ch	Reserved
280h - 298h	GPIO240-GPIO247 Pin Control Register
2ACh - 2BCh	GPIO253-GPIO257 Pin Control Register
2C0h	GPIO260 Pin Control Register
300h	Input GPIO[000:036]
304h	Input GPIO[040:076]
308h	Input GPIO[100:127]
30Ch	Input GPIO[140:176]
310h	Input GPIO[200:236]
314h	Input GPIO[240:276]
380h	Output GPIO[000:036]

TABLE 11-2: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
384h	Output GPIO[040:076]
388h	Output GPIO[100:127]
38Ch	Output GPIO[140:176]
390h	Output GPIO[200:236]
394h	Output GPIO[240:276]
500h - 51Ch	GPIO000-GPIO007 Pin Control Register 2
520h - 53Ch	GPIO010-GPIO017 Pin Control Register 2
540h - 55Ch	GPIO020-GPIO027 Pin Control Register 2
560h - 578h	GPIO030-GPIO036 Pin Control Register 2
580h - 59Ch	GPIO040-GPIO047 Pin Control Register 2
5A0h - 5BCh	GPIO050-GPIO057 Pin Control Register 2
5C0h - 5DCh	GPIO060-GPIO067 Pin Control Register 2
5E0h - 5F8h	GPIO070-GPIO076 Pin Control Register 2
600h - 61Ch	GPIO100-GPIO107 Pin Control Register 2
620h - 63Ch	GPIO110-GPIO117 Pin Control Register 2
640h - 65Ch	GPIO120-GPIO127 Pin Control Register 2
660h - 674h	GPIO130-GPIO135 Pin Control Register 2
680h - 69Ch	GPIO140-GPIO147 Pin Control Register 2
6A0h - 6BCh	GPIO150-GPIO157 Pin Control Register 2
6C0h - 6D8h	GPIO160-GPIO167 Pin Control Register 2
6E0h - 6F4h	GPIO170-GPIO175 Pin Control Register 2
700h - 71Ch	GPIO200-GPIO207 Pin Control Register 2
720h - 73Ch	GPIO210-GPIO217 Pin Control Register 2
740h - 75Ch	GPIO220-GPIO227 Pin Control Register 2
760h - 778h	Reserved
780h - 79Ch	GPIO240-GPIO247 Pin Control Register 2
7A0h - 7BCh	GPIO250-GPIO257 Pin Control Register 2
7C0h	GPIO260 Pin Control Register 2

11.6.1 PIN CONTROL REGISTERS

Two Pin Control Registers are implemented for each GPIO. The Pin Control Register format is described in Section 11.6.1.1, "Pin Control Register," on page 143. The Pin Control Register 2 format is described in Section 11.6.1.2, "Pin Control Register 2," on page 146. Pin Control Register address offsets and defaults for each product are defined in Section 11.5.3.1, "Pin Control Register Defaults," on page 140.

11.6.1.1 Pin Control Register

Offset	See Table 11-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	RESERVED	RES	-	-
24	GPIO input from pad On reads, Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [10]. **Note:** This bit is forced high when the selected power well is off as selected by the Power Gating Signal bits. See bits[3:2].	R	Note 11-1	RESET_SYS
23:17	RESERVED	RES	-	-
16	GPIO output data If enabled by the GPIO Output Control Select bit, the GPIO output data bit determines the level on the GPIO pin when the pin is configured for the GPIO output function. On writes: If enabled via the GPIO Output Control Select 0: GPIO[x] out = ‘0’ 1: GPIO[x] out = ‘1’ **Note:** If disabled via the GPIO Output Control Select then the GPIO[x] out pin is unaffected by writing this bit. On reads: Bit [16] returns the last programmed value, not the value on the pin.	R/W (GPIO Output Control Select = 0) R (GPIO Output Control Select=1)	Note 11-1	RESET_SYS
15	GPIO input disable This bit can be used to support undervoltage functionality. 1=disable input 0=do not disable input	R/W	Note 11-1	RESET_SYS
14	RESERVED	RES	-	-
13:12	Mux Control The Mux Control field determines the active signal function for a pin. 00 = GPIO Function Selected 01 = Signal Function 1 Selected 10 = Signal Function 2 Selected 11 = Signal Function 3 Selected	R/W	Note 11-1	RESET_SYS

Offset	See Table 11-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
11	Polarity 0 = Non-inverted 1 = Inverted When the Polarity bit is set to '1' and the Mux Control bits are greater than '00,' the selected signal function outputs are inverted and Interrupt Detection (int_det) sense defined in Table 11-3, "Edge Enable and Interrupt Detection Bits Definition" is inverted. When the Mux Control field selects the GPIO signal function (Mux = '00'), the Polarity bit does not effect the output. Regardless of the state of the Mux Control field and the Polarity bit, the state of the pin is always reported without inversion in the GPIO input register.	R/W	Note 11-1	RESET_SYS
10	GPIO Output Control Select Every GPIO has two mechanisms to set a GPIO data output: Output GPIO Bit located in the grouped GPIO Output Registers and the single GPIO output data bit located in bit 16 of this register. This control bit determines the source of the GPIO output. 0 = Pin Control Bit[16] GPIO output data bit enabled When this bit is zero the single GPIO output data bit is enabled. (GPIO output data is R/W capable and the Grouped Output GPIO is disabled (i.e., Read-Only). 1 = Grouped Output GPIO enable When this bit is one the GPIO output data write is disabled (i.e., Read-Only) and the Grouped Output GPIO is enabled (i.e., R/W). **Note:** See description in Section 11.5.1, "Accessing GPIOs".	R/W	Note 11-1	RESET_SYS
9	GPIO Direction 0 = Input 1 = Output The GPIO Direction bit controls the buffer direction only when the Mux Control field is '00' selecting the pin signal function to be GPIO. When the Mux Control field is greater than '00' (i.e., a non-GPIO signal function is selected) the GPIO Direction bit has no affect and the selected signal function logic directly controls the pin direction.	R/W	Note 11-1	RESET_SYS
8	Output Buffer Type 0 = Push-Pull 1 = Open Drain **Note:** Unless explicitly stated otherwise, pins with (I/O/OD) or (O/OD) in their buffer type column in the tables in are compliant with the following Programmable OD/PP Multiplexing Design Rule: Each compliant pin has a programmable open drain/push-pull buffer controlled by the Output Buffer Type bit in the associated Pin Control Register. The state of this bit controls the mode of the interface buffer for all selected functions, including the GPIO function.	R/W	Note 11-1	RESET_SYS

Offset	See Table 11-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Edge Enable (edge_en) 0 = Edge detection disabled 1 = Edge detection enabled **Note:** See Table 11-3, "Edge Enable and Interrupt Detection Bits Definition".	R/W	Note 11-1	RESET_SYS
6:4	Interrupt Detection (int_det) The interrupt detection bits determine the event that generates a GPIO_Event. **Note:** See Table 11-3, "Edge Enable and Interrupt Detection Bits Definition". **Note:** Since the GPIO input is always available, even when the GPIO is not selected as the alternate function, the GPIO interrupts may be used for detecting pin activity on alternate functions. The only exception to this is the analog functions (e.g., ADC inputs)	R/W	Note 11-1	RESET_SYS
3:2	Power Gating Signals (PGS) The Power Gating Signals provide the chip Power Emulation options. The pin will be tristated when the selected power well is off (i.e., gated) as indicated. The Emulated Power Well column defined in Pin Multiplexing indicates the emulation options supported for each signal. The Signal Power Well column defines the buffer power supply per function. **Note:** Note that all GPIOs support Power Gating unless otherwise noted. 00 = VTR The output buffer is tristated when VTR_PWRGD = 0. 01 = 10 = Unpowered. The always unpowered setting on a GPIO will force the pin to tristate. The input and output are disabled, and the pad is in the lowest power state. 11 = Reserved **Note:** VBAT Powered Signals are always powered by the VBAT rail and power well emulation does not apply. For VBAT powered signals this field should be set to 00.	R/W	Note 11-1	RESET_SYS
1:0	PU/PD (PU_PD) These bits are used to enable an internal pull-up or pull-down resistor device on the pin. 00 = None. Pin tristates when no active driver is present on the pin. 01 = Pull Up Enabled 10 = Pull Down Enabled 11 = Repeater mode. Pin is kept at previous voltage level when no active driver is present on the pin.	R/W	Note 11-1	RESET_SYS

Note 11-1 See Section 3.5, "GPIO Register Assignments" for the default values and Table 11-2, "Register Summary" and Table 3.6, "Register Map" for register offset value for each GPIO Pin Control Register.

Note 11-2 Repeater mode is not available on over voltage protected pins.

TABLE 11-3: EDGE ENABLE AND INTERRUPT DETECTION BITS DEFINITION

Edge Enable	Interrupt Detection Bits			Selected Function
D7	D6	D5	D4	
0	0	0	0	Low Level Sensitive
0	0	0	1	High Level Sensitive
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Interrupt events are disabled
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	1	0	1	Rising Edge Triggered
1	1	1	0	Falling Edge Triggered
1	1	1	1	Either Edge Triggered

Note: Only edge triggered interrupts can wake up the main ring oscillator. The GPIO must be enabled for edge-triggered interrupts and the GPIO interrupt must be enabled in the interrupt aggregator in order to wake up the ring when the ring is shut down.

APPLICATION NOTE:

1. All GPIO interrupt detection configurations default to '0000', which is low level interrupt. Having interrupt detection enabled will un-gated the clock to the GPIO module whenever the interrupt is active, which increases power consumption. Interrupt detection should be disabled when not required to save power.
2. Changing the configuration of the Interrupt edge and detection bits may generate an interrupt if it is enabled. The GPIO should be configured and associated status bits should be cleared before enabling the Interrupt.

11.6.1.2 Pin Control Register 2

Offset	See Note 11-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5:4	DRIVE_STRENGTH These bits are used to select the drive strength on the pin. 00 = 2mA 01 = 4mA 10 = 8mA 11 = 12mA	R/W	00	RESET_SYS

Offset	See Note 11-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3:1	Reserved	RES	-	-
0	Slew Rate This bit is used to select the slew rate on the pin 1= fast (Should be set to 1 always) 0= slow	R/W	0h	RESET_SYS

11.6.2 GPIO OUTPUT REGISTERS

If enabled by the GPIO Output Control Select bit, the grouped GPIO Output bits determine the level on the GPIO pin when the pin is configured for the GPIO output function.

On writes:

If enabled via the GPIO Output Control Select

0: GPIO[x] out = ‘0’

1: GPIO[x] out = ‘1’

If disabled via the GPIO Output Control Select then the GPIO[x] out pin is unaffected by writing the corresponding GPIO bit in the grouped Output GPIO[xxx:yyy] register.

On reads:

The GPIO output bit in the grouped Output GPIO[xxx:yyy] register returns the last programmed value, not the value on the pin.

11.6.2.1 Output GPIO[000:036]

Offset	380h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[036:030] Output	R/W	00h	RESET_SYS
23:16	GPIO[027:020] Output	R/W	00h	RESET_SYS
15:8	GPIO[017:010] Output	R/W	00h	RESET_SYS
7:0	GPIO[007:000] Output	R/W	00h	RESET_SYS

11.6.2.2 Output GPIO[040:076]

Offset	384h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	RESERVED	RES	-	-
30:24	GPIO[076:070] Output	R/W	00h	RESET_SYS
23:16	GPIO[067:060] Output	R/W	00h	RESET_SYS
15:8	GPIO[057:050] Output	R/W	00h	RESET_SYS
7:0	GPIO[047:040] Output	R/W	00h	RESET_SYS

11.6.2.3 Output GPIO[100:127]

Offset	388h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[136:130] Output	R/W	00h	RESET_SYS
23:16	GPIO[127:120] Output	R/W	00h	RESET_SYS
15:8	GPIO[117:110] Output	R/W	00h	RESET_SYS
7:0	GPIO[107:100] Output	R/W	00h	RESET_SYS

11.6.2.4 Output GPIO[140:176]

Offset	38Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[176:170] Output	R/W	00h	RESET_SYS

Offset	38Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
23:16	GPIO[167:160] Output	R/W	00h	RESET_SYS
15:8	GPIO[157:150] Output	R/W	00h	RESET_SYS
7:0	GPIO[147:140] Output	R/W	00h	RESET_SYS

11.6.2.5 Output GPIO[200:236]

Offset	390h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[236:230] Output	R/W	00h	RESET_SYS
23:16	GPIO[227:220] Output	R/W	00h	RESET_SYS
15:8	GPIO[217:210] Output	R/W	00h	RESET_SYS
7:0	GPIO[207:200] Output	R/W	00h	RESET_SYS

11.6.2.6 Output GPIO[240:276]

Offset	394h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[276:270] Output	R/W	00h	RESET_SYS
23:16	GPIO[267:260] Output	R/W	00h	RESET_SYS

Offset	394h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:8	GPIO[257:250] Output	R/W	00h	RESET_SYS
7:0	GPIO[247:240] Output	R/W	00h	RESET_SYS

11.6.3 GPIO INPUT REGISTERS

The GPIO Input Registers can always be used to read the state of a pin, even when the pin is in an output mode and/or when a signal function other than the GPIO signal function is selected; i.e., the Pin Control Register Mux Control bits are not equal to ‘00.’

The MSbit of the Input GPIO registers have been implemented as a read/write scratch pad bit to support processor specific instructions.

Note: Bits associated with GPIOs that are not implemented are shown as Reserved.

11.6.3.1 Input GPIO[000:036]

Offset	300h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[036:030] Input	R	00h	RESET_SYS
23:16	GPIO[027:020] Input	R	00h	RESET_SYS
15:8	GPIO[017:010] Input	R	00h	RESET_SYS
7:0	GPIO[007:000] Input	R	00h	RESET_SYS

11.6.3.2 Input GPIO[040:076]

Offset	304h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[076:070] Input	R	00h	RESET_SYS
23:16	GPIO[067:060] Input	R	00h	RESET_SYS
15:8	GPIO[057:050] Input	R	00h	RESET_SYS
7:0	GPIO[047:040] Input	R	00h	RESET_SYS

11.6.3.3 Input GPIO[100:127]

Offset	308h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[136:130] Input	R	00h	RESET_SYS
23:16	GPIO[127:120] Input	R	00h	RESET_SYS
15:8	GPIO[117:110] Input	R	00h	RESET_SYS
7:0	GPIO[107:100] Input	R	00h	RESET_SYS

11.6.3.4 Input GPIO[140:176]

Offset	30Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:16	GPIO[176:160] Input	R	00h	RESET_SYS
15:8	GPIO[157:150] Input	R	00h	RESET_SYS
7:0	GPIO[147:140] Input	R	00h	RESET_SYS

11.6.3.5 Input GPIO[200:236]

Offset	310h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[236:230] Input	R	00h	RESET_SYS
23:16	GPIO[227:220] Input	R	00h	RESET_SYS
15:8	GPIO[217:210] Input	R	00h	RESET_SYS
7:0	GPIO[207:200] Input	R	00h	RESET_SYS

11.6.3.6 Input GPIO[240:276]

Offset	314h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[276:270] Input	R	00h	RESET_SYS

Offset	314h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
23:16	GPIO[267:260] Input	R	00h	RESET_SYS
15:8	GPIO[257:250] Input	R	00h	RESET_SYS
7:0	GPIO[247:240] Input	R	00h	RESET_SYS

12.0 WATCHDOG TIMER (WDT)

12.1 Introduction

The function of the Watchdog Timer is to provide a mechanism to detect if the internal embedded controller has failed.

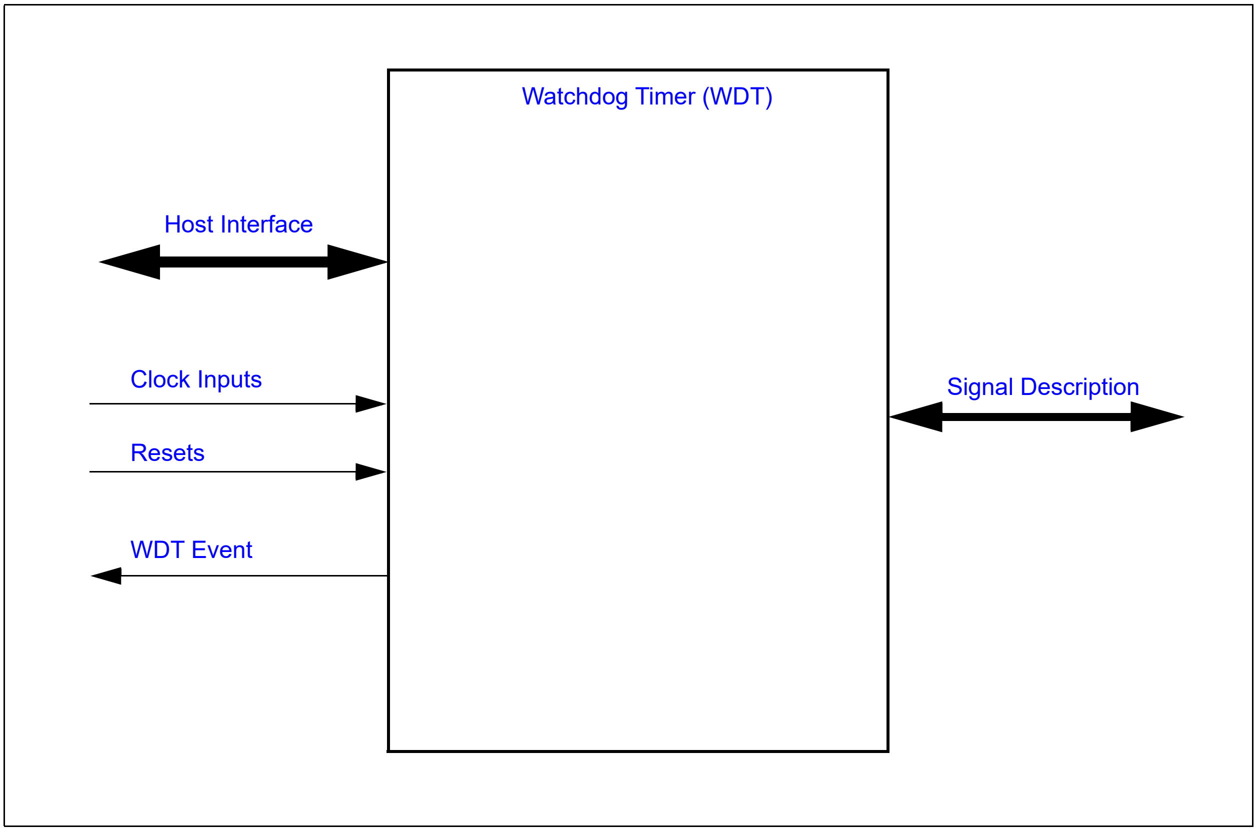
When enabled, the Watchdog Timer (WDT) circuit will generate a WDT Event if the user program fails to reload the WDT within a specified length of time known as the WDT Interval.

12.2 Interface

This block is designed to be accessed internally via a registered host interface

12.3 Host Interface

FIGURE 12-1: I/O DIAGRAM OF BLOCK



The registers defined for the Watchdog Timer (WDT) are accessible by the embedded controller as indicated in Section 12.7, "EC Registers". All registers accesses are synchronized to the host clock and complete immediately. Register reads/writes are not delayed by the 32KHz.

12.4 Signal Description

12.4.1 SIGNAL INTERFACE

There are no external signals for this block.

12.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

12.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block reside on this single power well.

12.5.2 CLOCK INPUTS

Name	Description
32KHz	The 32KHz clock input is the clock source to the Watchdog Timer functional logic, including the counter.

12.5.3 RESETS

TABLE 12-1: RESET INPUTS

Name	Description
RESET_SYS	Power on Reset to the block. This signal resets all the register and logic in this block to its default state following a POR or a WDT Event event.
RESET_SYS_nWDT	This reset signal is used on WDT registers/bits that need to be preserved through a WDT Event.

TABLE 12-2: RESET OUTPUTS

Source	Description
WDT Event	Pulse generated when WDT expires. This signal is used to either generate interrupt WDT_INT, if WDT Control Register bit 9 is set to 1b (WDT_INT_ENABLE), or reset the embedded controller and its subsystem, if WDT Control Register bit 9 is set to 0b. The event is cleared after a RESET_SYS.

12.6 Description

12.6.1 WDT OPERATION

12.6.1.1 WDT Activation Mechanism

The WDT is activated by the following sequence of operations during normal operation:

1. Load the WDT Load Register with the count value.
2. Set the WDT_ENABLE bit in the WDT Control Register.

The WDT Activation Mechanism starts the WDT decrementing counter.

12.6.1.2 WDT Deactivation Mechanism

The WDT is deactivated by the clearing the WDT_ENABLE bit in the WDT Control Register. The WDT Deactivation Mechanism places the WDT in a low power state in which clock are gated and the counter stops decrementing.

12.6.1.3 WDT Reload Mechanism

The WDT must be reloaded within periods that are shorter than the programmed watchdog interval; otherwise, the WDT will underflow and a WDT Event will be generated and the WDT bit in Power-Fail and Reset Status Register on page 290 will be set. It is the responsibility of the user program to continually execute code which reloads the watchdog timer, causing the counter to be reloaded.

There are three methods of reloading the WDT: a write to the WDT Load Register, a write to the WDT Kick Register, or WDT event.

12.6.1.4 WDT Interval

The WDT Interval is the time it takes for the WDT to decrements from the WDT Load Register value to 0000h. The WDT Count Register value takes 33/32KHz seconds (ex. 33/32.768 KHz = 1.007ms) to decrement by 1 count.

12.6.1.5 WDT STALL Operation

There are three STALL_ENABLE control bits in the WDT Control Register. If enabled, and the STALL event is asserted, the WDT stops decrementing, and the WDT enters a low power state. When a WDT STALL event is de-asserted, the counter continues decrementing from the value it had when the STALL was asserted.

12.7 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Watchdog Timer (WDT) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 12-3: REGISTER SUMMARY

Offset	Register Name
00h	WDT Load Register
04h	WDT Control Register
08h	WDT Kick Register
0Ch	WDT Count Register
10h	WDT Status Register
14h	WDT Int Enable Register

12.7.1 WDT LOAD REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	WDT_LOAD Writing this field reloads the Watch Dog Timer counter.	R/W	FFFFh	RESET_SYS

12.7.2 WDT CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	RES	-	-
9	WDT_RESET If the WDT_RESET bit is set and the watch dog timer expires, the Watch dog module will generate interrupt and the WDT_RESET bit will be cleared. If this bit is not set, when the watch dog timer expires EC and its sub-system is reset.	R/W	0b	RESET_SYS
8:5	Reserved	RES	-	-
4	JTAG_STALL This bit enables the WDT Stall function if JTAG or SWD debug functions are active 1=The WDT is stalled while either JTAG or SWD is active 0=The WDT is not affected by the JTAG debug interface	R/W	0b	RESET_SYS
3	WEEK_TIMER_STALL This bit enables the WDT Stall function if the Week Timer is active. 1=The WDT is stalled while the Week Timer is active 0=The WDT is not affected by the Week Timer	R/W	0b	RESET_SYS
2	HIBERNATION_TIMER_STALL This bit enables the WDT Stall function if the Hibernation Timer 0 or Hibernation Timer 1 is active. 1=The WDT is stalled while the Hibernation Timer 0 is active 0=The WDT is not affected by Hibernation Timer 0	R/W	0b	RESET_SYS
1	TEST	R	0b	RESET_SYS
0	WDT_ENABLE In WDT Operation, the WDT is activated by the sequence of operations defined in Section 12.6.1.1, "WDT Activation Mechanism" and deactivated by the sequence of operations defined in Section 12.6.1.2, "WDT Deactivation Mechanism". 1=block enabled 0=block disabled	R/W	0b	RESET_SYS

12.7.3 WDT KICK REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	KICK The WDT Kick Register is a strobe. Reads of this register return 0. Writes to this register cause the WDT to reload the WDT Load Register value and start decrementing when the WDT_ENABLE bit in the WDT Control Register is set to ‘1’. When the WDT_ENABLE bit in the WDT Control Register is cleared to ‘0’, writes to the WDT Kick Register have no effect.	W	n/a	RESET_SYS

12.7.4 WDT COUNT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	WDT_COUNT This read-only register provide the current WDT count.	R	FFFFh	RESET_ SYS

12.7.5 WDT STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	WDT_EVENT_IRQ This bit indicates the status of interrupt from Watch dog module.	R/W1C	0h	RESET_ SYS

12.7.6 WDT INT ENABLE REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	WDT_INT_ENABLE This is the interrupt enables bit for WDT_INT interrupt. 1b - WDT_INT Interrupt Enable 0b - WDT_INT Interrupt Disabled	R/W	0h	RESET_ SYS

13.0 16/32 BIT BASIC TIMER

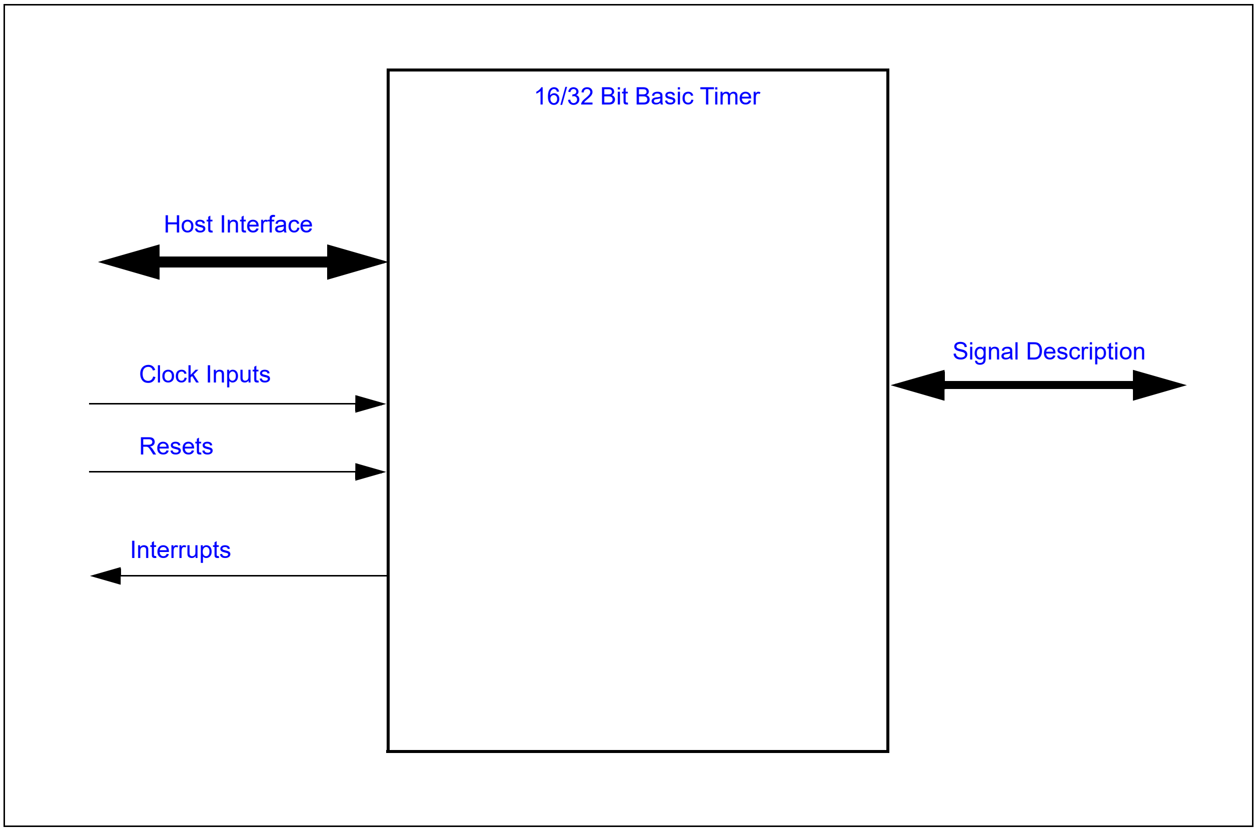
13.1 Introduction

This timer block offers a simple mechanism for firmware to maintain a time base. This timer may be instantiated as 16 bits or 32 bits. The name of the timer instance indicates the size of the timer.

13.2 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 13-1: I/O DIAGRAM OF BLOCK



13.3 Signal Description

There are no external signals for this block.

13.4 Host Interface

The Embedded Controller (EC) may access this block via the registers defined in Section 13.9, "EC-Only Registers," on page 162.

13.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

13.5.1 POWER DOMAINS

TABLE 13-1: POWER SOURCES

Name	Description
VTR_CORE	The timer control logic and registers are all implemented on this single power domain.

13.5.2 CLOCK INPUTS

TABLE 13-2: CLOCK INPUTS

Name	Description
48MHz	This is the clock source to the timer logic. The Pre-scaler may be used to adjust the minimum resolution per bit of the counter.

13.5.3 RESETS

TABLE 13-3: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.
SOFT_RESET	This reset signal, which is created by this block, resets all the logic and registers to their initial default state. This reset is generated by the block when the SOFT_RESET bit is set in the Timer Control Register register.
Timer_Reset	This reset signal, which is created by this block, is asserted when either the RESET_SYS or the SOFT_RESET signal is asserted. The RESET_SYS and SOFT_RESET signals are OR'd together to create this signal.

13.6 Interrupts

TABLE 13-4: EC INTERRUPTS

Source	Description
TIMER_16_*x*	This interrupt event fires when a 16-bit timer *x* reaches its limit. This event is sourced by the EVENT_INTERRUPT status bit if enabled.
TIMER_32_*x*	This interrupt event fires when a 32-bit timer *x* reaches its limit. This event is sourced by the EVENT_INTERRUPT status bit if enabled.
Note: x represents the instance number.	

13.7 Low Power Modes

The Basic Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active.

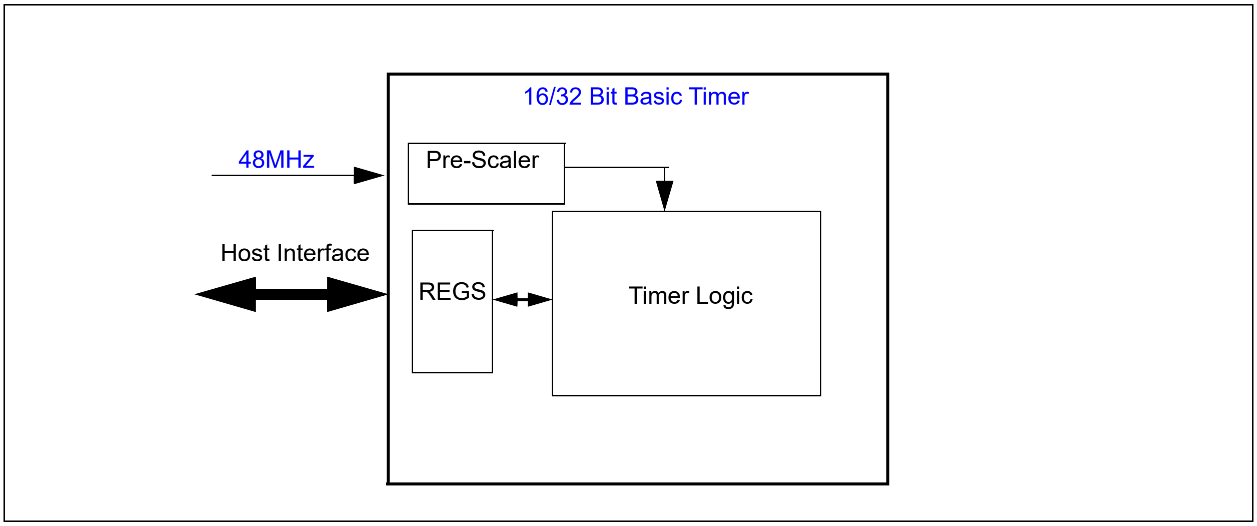
The sleep state of this timer is as follows:

- Asleep while the block is not Enabled
- Asleep while the block is not running (start inactive).
- Asleep while the block is halted (even if running).

The block is active while start is active.

13.8 Description

FIGURE 13-2: BLOCK DIAGRAM



This timer block offers a simple mechanism for firmware to maintain a time base in the design. The timer may be enabled to execute the following features:

- Programmable resolution per LSB of the counter via the Pre-scale bits in the Timer Control Register
- Programmable as either an up or down counter
- One-shot or Continuous Modes
- In one-shot mode the Auto Restart feature stops the counter when it reaches its limit and generates a level event.
- In Continuous Mode the Auto Restart feature restarts that counter from the programmed preload value and generates a pulse event.
- Counter may be reloaded, halted, or started via the Timer Control register
- Block may be reset by either a Power On Reset (POR) or via a Soft Reset.

13.9 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Basic Timer. The addresses of each register listed in this table are defined as a relative offset to the “Base Address” of that instance, defined in the Device Inventory chapter and will follow the instance naming as listed in **TABLE 13-5: “CEC1712 Instance Naming Convention”**.

TABLE 13-5: CEC1712 INSTANCE NAMING CONVENTION

Block Instance	Host
16-Bit Basic Timer x	EC
32-Bit Basic Timer x	EC
Note: x represents the instance number.	

TABLE 13-6: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Timer Count Register
04h	Timer Preload Register
08h	Timer Status Register
0Ch	Timer Int Enable Register
10h	Timer Control Register

13.9.1 TIMER COUNT REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER This is the value of the Timer counter. This is updated by Hardware but may be set by Firmware. If it is set while the Hardware Timer is operating, functionality can not be guaranteed. When read, it is buffered so single byte reads will be able to catch the full 4 byte register without it changing. - For 16 bit Basic Timer, bits 0 to 15 are r/w counter bits. Bits 31 down to 16 are reserved. Reads of bits 31 down to 16 return 0 and writes have no effect. - For 32 bit Basic Timer, bits 0 to 31 are r/w counter bits.	R/W	0h	Timer_Reset

13.9.2 TIMER PRELOAD REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PRE_LOAD This is the value of the Timer pre-load for the counter. This is used by H/W when the counter is to be restarted automatically; this will become the new value of the counter upon restart. The size of the Pre-Load value is the same as the size of the counter. - For 16 bit Basic Timer, bits 0 to 15 are r/w pre-load bits. Bits 31 down to 16 are reserved. Reads of bits 31 down to 16 return 0 and writes have no effect. - For 32 bit Basic Timer, bits 0 to 31 are r/w pre-load bits.	R/W	0h	Timer_Reset

13.9.3 TIMER STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	RES	-	-
0	EVENT_INTERRUPT This is the interrupt status that fires when the timer reaches its limit. This may be level or a self clearing signal cycle pulse, based on the AUTO_RESTART bit in the Timer Control Register. If the timer is set to automatically restart, it will provide a pulse, otherwise a level is provided.	R/WC	0h	Timer_Reset

13.9.4 TIMER INT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	RES	-	-
0	EVENT_INTERRUPT_ENABLE This is the interrupt enable for the status EVENT_INTERRUPT bit in the Timer Status Register	R/W	0h	Timer_Reset

13.9.5 TIMER CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	PRE_SCALE This is used to divide down the system clock through clock enables to lower the power consumption of the block and allow slow timers. Updating this value during operation may result in erroneous clock enable pulses until the clock divider restarts. The number of clocks per clock enable pulse is (Value + 1); a setting of 0 runs at the full clock speed, while a setting of 1 runs at half speed.	R/W	0h	Timer_Reset
15:8	Reserved	RES	-	-
7	HALT This is a halt bit. This will halt the timer as long as it is active. Once the halt is inactive, the timer will start from where it left off. 1=Timer is halted. It stops counting. The clock divider will also be reset. 0=Timer runs normally	R/W	0h	Timer_Reset
6	RELOAD This bit reloads the counter without interrupting it operation. This will not function if the timer has already completed (when the START bit in this register is '0'). This is used to periodically prevent the timer from firing when an event occurs. Usage while the timer is off may result in erroneous behavior.	R/W	0h	Timer_Reset
5	START This bit triggers the timer counter. The counter will operate until it hits its terminating condition. This will clear this bit. It should be noted that when operating in restart mode, there is no terminating condition for the counter, so this bit will never clear. Clearing this bit will halt the timer counter. Setting this bit will: • Reset the clock divider counter. • Enable the clock divider counter. • Start the timer counter. • Clear all interrupts. Clearing this bit will: • Disable the clock divider counter. • Stop the timer counter.	R/W	0h	Timer_Reset
4	SOFT_RESET This is a soft reset. This is self clearing 1 cycle after it is written.	WO	0h	Timer_Reset
3	AUTO_RESTART This will select the action taken upon completing a count. 1=The counter will automatically restart the count, using the contents of the Timer Preload Register to load the Timer Count Register The interrupt will be set in edge mode 0=The counter will simply enter a done state and wait for further control inputs. The interrupt will be set in level mode.	R/W	0h	Timer_Reset

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	COUNT_UP This selects the counter direction. When the counter in incrementing the counter will saturate and trigger the event when it reaches all F's. When the counter is decrementing the counter will saturate when it reaches 0h. 1=The counter will increment 0=The counter will decrement	R/W	0h	Timer_Reset
1	Reserved	RES	-	-
0	ENABLE This enables the block for operation. 1=This block will function normally 0=This block will gate its clock and go into its lowest power state	R/W	0h	Timer_Reset

14.0 INPUT CAPTURE AND COMPARE TIMER

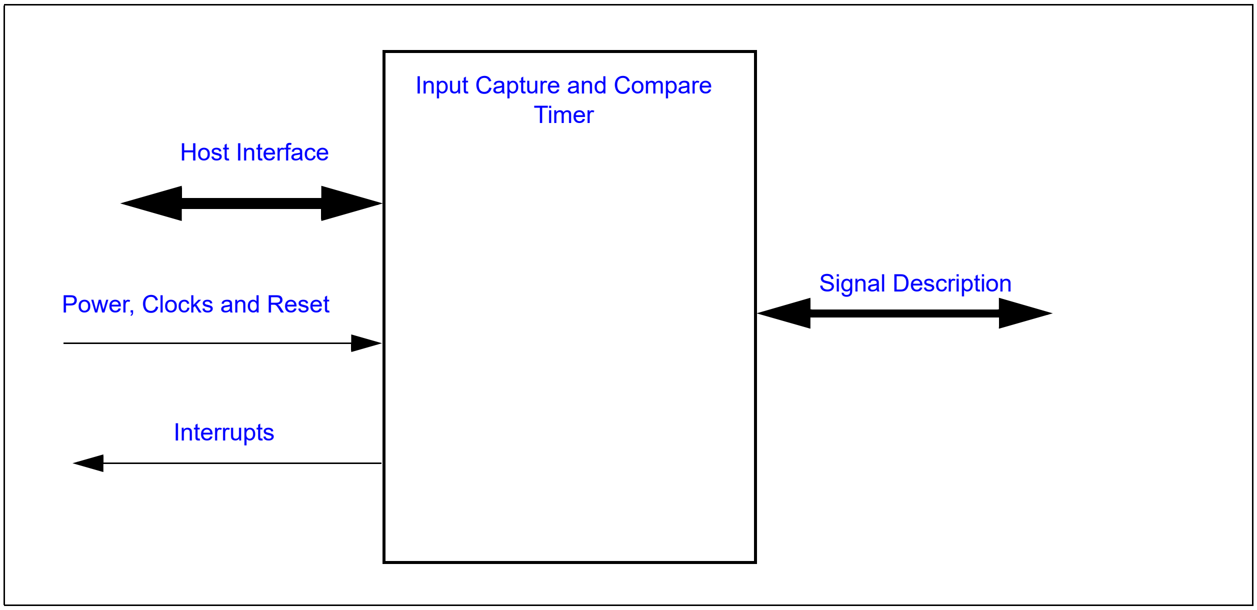
14.1 Introduction

The Input Capture and Compare Timers block contains a 32-bit timer running at the main system clock frequency. The timer is free-running and is associated with six 32-bit capture registers and two compare registers. Each capture register can record the value of the free-running timer based on a programmable edge of its associated input pin. An interrupt can be generated for each capture register each time it acquires a new timer value. The timer can also generate an interrupt when it automatically resets and can additionally generate two more interrupts when the timer matches the value in either of two 32-bit compare registers.

14.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 14-1: I/O DIAGRAM OF BLOCK



14.3 Signal Description

TABLE 14-1: SIGNAL DESCRIPTION

Name	Direction	Description
ICTx	INPUT	External capture trigger signal for Capture Register.
CTOUT0	OUTPUT	External compare match signal for Compare Register 0
CTOUT1	OUTPUT	External compare match signal for Compare Register 1
Note: Any ICTx can be connected to any Capture register using the ICT MUX Select Register.		

14.4 Host Interface

The registers defined for 16-bit Timers are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

14.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

14.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

14.5.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for this block.

14.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

14.6 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
CAPTURE TIMER	This interrupt event fires when the 32-bit free running counter overflows from FFFF_FFFFh to 0000_0000h.
CAPTURE 0	This interrupt event fires when Capture Register 0 acquires a new value.
CAPTURE 1	This interrupt event fires when Capture Register 1 acquires a new value.
CAPTURE 2	This interrupt event fires when Capture Register 2 acquires a new value.
CAPTURE 3	This interrupt event fires when Capture Register 3 acquires a new value.
CAPTURE 4	This interrupt event fires when Capture Register 4 acquires a new value.
CAPTURE 5	This interrupt event fires when Capture Register 5 acquires a new value.
COMPARE 0	This interrupt event fires when the contents of Compare 0 Register match the contents of the Free Running Counter.
COMPARE 1	This interrupt event fires when the contents of Compare 1 Register match the contents of the Free Running Counter.

14.7 Low Power Modes

The Capture and Compare Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active. The block is inactive if the ACTIVATE bit is de-asserted, and will also become inactive when the block's SLEEP_EN signal is asserted.

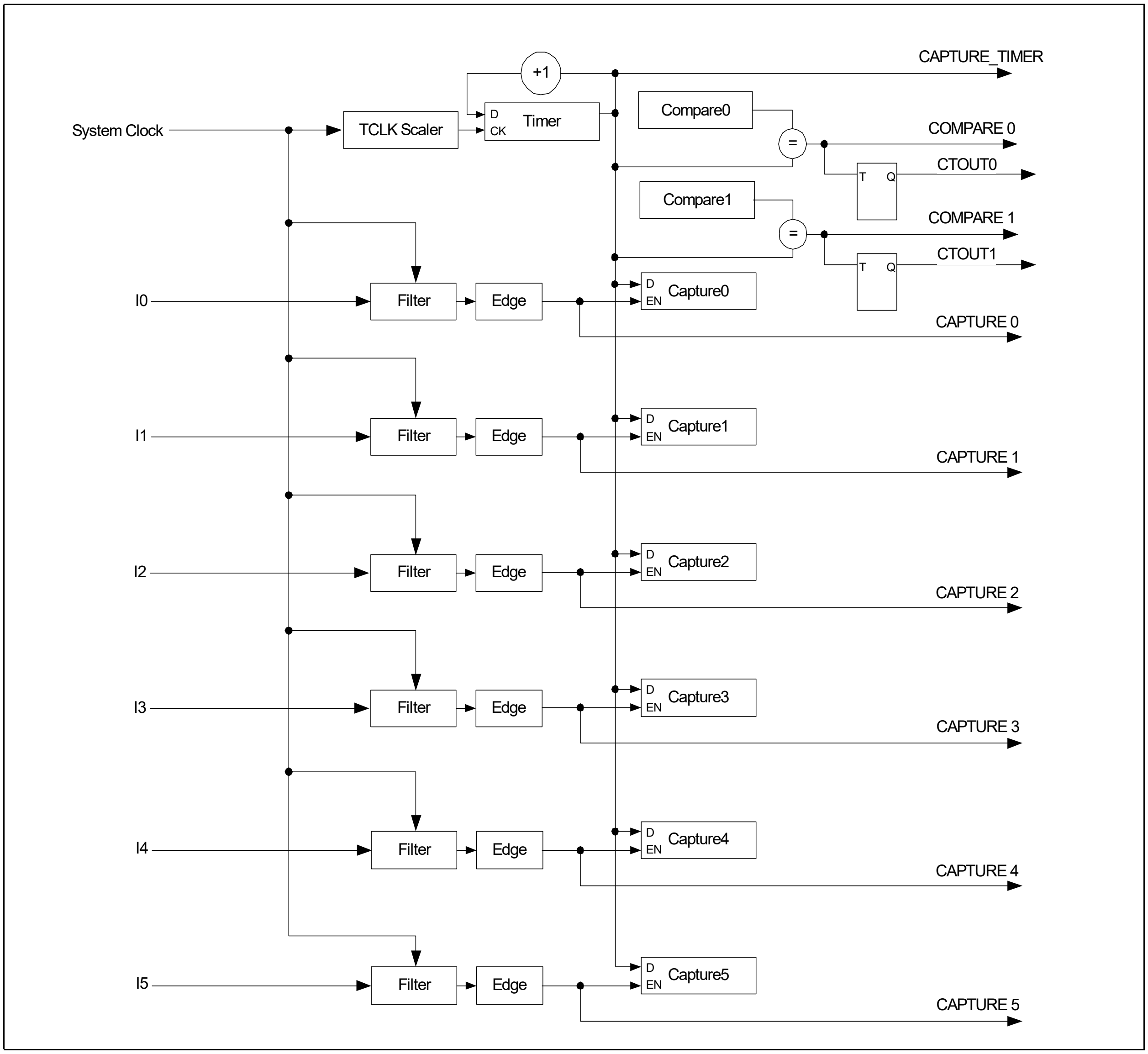
When the block returns from sleep, if enabled, the Free Running Timer Register value will continue counting from where it was when the block entered the Sleep state.

14.8 Description

The Input Capture and Compare Timer block has ICT Channels inputs and these can be connected to any of the 6 Capture Compare timer

Note: The CCT0 to CCT5 blocks are expanded and shown in **FIGURE 14-2: "Capture and Compare Timer Block Diagram"**

FIGURE 14-2: CAPTURE AND COMPARE TIMER BLOCK DIAGRAM



14.8.1 TIMER CLOCK

Any of the frequencies listed in Table 14-2 may be used as the time base for the Free Running Counter.

TABLE 14-2: TIMER CLOCK FREQUENCIES

Timer Clock Select	Frequency Divide Select	Frequency Selected
0000b	Divide by 1	48MHz
0001b	Divide by 2	24MHz
0010b	Divide by 4	12MHz
0011b	Divide by 8	6MHz
0100b	Divide by 16	3MHz
0101b	Divide by 32	1.5MHz
0110b	Divide by 64	750KHz
0111b	Divide by 128	375KHz
1xxxb	Reserved	Reserved

For the Timer Clock, the **Timer Clock Select** value is defined by the TCLK field in the Capture and Compare Timer Control Register

14.8.2 FILTER CLOCK AND NOISE FILTER

The noise filter uses the Filter Clock (FCLK) to filter the signal on the Input Capture pins. An Input Capture pin must remain in the same state for three FCLK ticks before the internal state changes. The FILTER_BYPASS bit for the Input Capture pin may be used to bypass the input filter. Each Capture Register can individually bypass the filter.

When the input filter is bypassed, the minimum period of FCLK must be at least 2X the duration of an input signal pulse in order for an edge event to be captured reliably. When the input filter is enabled, the minimum period of FCLK must be at least 4X the duration of an input signal pulse in order for an edge event to be captured reliably.

14.9 Operation

14.9.1 INPUT CAPTURE

The Input Capture block consists of a free-running 32-bit timer and 2 capture registers. Each of the capture registers is associated with an input pin as well as an interrupt source bit in the Interrupt Aggregator: The Capture registers store the current value of the Free Running timer whenever the associated input signal changes, according to the programmed edge detection. An interrupt is also generated to the EC. The Capture registers are read-only. The registers are updated every time an edge is detected. If software does not read the register before the next edge, the value is lost.

14.9.2 COMPARE TIMER

There are two 32-bit Compare registers. Each of these registers can independently generate an interrupt to the EC when the 32-bit Free Running Timer matches the contents of the Compare register. The compare operation for each is enabled or disabled by a bit in the Capture and Compare Timer Control Register.

14.9.2.1 Interrupt Generation

Whenever a Compare Timer is enabled and the Compare register matches the Free Running Timer, a COMPARE event is sent to the Interrupt Aggregator. The event will trigger an EC interrupt if enabled by the appropriate Interrupt Enable register in the Aggregator.

14.9.2.2 Compare Output Generation

Each Compare Timer is associated with a toggle flip-flop. When the 32-bit Free Running Timer matches the contents of the Compare register the output off the flip-flop is complemented. Each of the toggle flip-flops can be independently set or cleared by using the COMPARE_SET or COMPARE_CLEAR fields, respectively, in the Capture and Compare Timer Control Register.

A Compare Timer should be disabled before setting or clearing the output, when updating the Compare register, or when updating the Free Running Timer, so spurious events are not generated by the matcher.

14.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Input Capture and Compare Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Note: All registers in this block must be accessed as DWORDs.

TABLE 14-3: REGISTER SUMMARY

Offset	Register Name
00h	Capture and Compare Timer Control Register
04h	Capture Control 0 Register
08h	Capture Control 1 Register
0Ch	Free Running Timer Register
10h	Capture 0 Register
14h	Capture 1 Register
18h	Capture 2 Register
1Ch	Capture 3 Register
20h	Capture 4 Register
24h	Capture 5 Register
28h	Compare 0 Register
2Ch	Compare 1 Register
30h	ICT MUX Select Register

14.10.1 CAPTURE AND COMPARE TIMER CONTROL REGISTER

Note: It is not recommended to use Read-Modify-Write operations on this register. May inadvertently cause the COMPARE_SET and COMPARE_CLEAR bits to be written to ‘1’ in error.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	RES	-	-
25	COMPARE_CLEAR0 When read, returns the current value off the Compare Timer Output 0 state. If written with a ‘1b’, the output state is cleared to ‘0’. Writes have no effect if COMPARE_SET1 in this register is written with a ‘1b’ at the same time. Writes of ‘0b’ have no effect.	R/WC	0	RESET_SYS

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
24	COMPARE_CLEAR1 When read, returns the current value off the Compare Timer Output 1 state. If written with a ‘1b’, the output state is cleared to ‘0’. Writes have no effect if COMPARE_SET0 in this register is written with a ‘1b’ at the same time. Writes of ‘0b’ have no effect.	R/WC	0	RESET _SYS
23:18	Reserved	RES	-	-
17	COMPARE_SET0 When read, returns the current value off the Compare Timer Output 0 state. • If written with a ‘1b’, the output state is set to ‘1’. • Writes of ‘0b’ have no effect	R/WS	0	RESET _SYS
16	COMPARE_SET1 When read, returns the current value off the Compare Timer Output 1 state. If written with a ‘1b’, the output state is set to ‘1’. Writes of ‘0b’ have no effect	R/WS	0	RESET _SYS
15:10	Reserved	RES	-	-
9	COMPARE_ENABLE1 Compare Enable for Compare 1 Register. When enabled, a match between the Compare 1 Register and the Free Running Timer Register will cause the TOUT1 output to toggle and will send a COMPARE event to the Interrupt Aggregator. 1=Enabled 0=Disabled	R/W	0b	RESET _SYS
8	COMPARE_ENABLE0 Compare Enable for Compare 0 Register. When enabled, a match between the Compare 0 Register and the Free Running Timer Register will cause the TOUT0 output to toggle and will send a COMPARE event to the Interrupt Aggregator. 1=Enabled 0=Disabled	R/W	0b	RESET _SYS
7	Reserved	RES	-	-
6:4	TCLK This 3-bit field sets the clock source for the Free-Running Counter. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET _SYS
3	Reserved	RES	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	FREE_RESET Free Running Timer Reset. This bit stops the timer and resets the internal counter to 0000_0000h. This bit does not affect the FREE_ENABLE bit. This bit is self clearing after the timer is reset. 1=Timer reset 0=Normal timer operation	R/W	0h	RESET_SYS
1	FREE_ENABLE Free-Running Timer Enable. This bit is used to start and stop the free running timer. This bit does not reset the timer count. The timer starts counting at 0000_0000h on reset and wraps around back to 0000_0000h after it reaches FFFF_FFFFh. The FREE_ENABLE bit is cleared after the RESET cycle is done. Firmware must poll the FREE_RESET bit to determine when it is safe to re-enable the timer. 1=Timer is enabled. The Free Running Timer Register is read-only. 0=Timer is disabled. The Free Running Timer Register is writable.	R/W	0h	RESET_SYS
0	ACTIVATE 1=The timer block is in a running state 0=The timer block is powered down and all clocks are gated	R/W	0h	RESET_SYS

14.10.2 CAPTURE CONTROL 0 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	FCLK_SEL3 This 3-bit field sets the clock source for the input filter for Capture Register 3. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
28:27	Reserved	RES	-	-
26	FILTER_BYP3 This bit enables bypassing the input noise filter for Capture Register 3, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
25:24	CAPTURE_EDGE3 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 3. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
23:21	FCLK_SEL2 This 3-bit field sets the clock source for the input filter for Capture Register 2. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
20:19	Reserved	RES	-	-
18	FILTER_BYP2 This bit enables bypassing the input noise filter for Capture Register 2, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
17:16	CAPTURE_EDGE2 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 2. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
15:13	FCLK_SEL1 This 3-bit field sets the clock source for the input filter for Capture Register 1. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
12:11	Reserved	RES	-	-
10	FILTER_BYP1 This bit enables bypassing the input noise filter for Capture Register 1, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
9:8	CAPTURE_EDGE1 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 1. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	FCLK_SEL0 This 3-bit field sets the clock source for the input filter for Capture Register 0. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
4:3	Reserved	RES	-	-
2	FILTER_BYP0 This bit enables bypassing the input noise filter for Capture Register 0, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
1:0	CAPTURE_EDGE0 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 0. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS

14.10.3 CAPTURE CONTROL 1 REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:13	FCLK_SEL5 This 3-bit field sets the clock source for the input filter for Capture Register 5. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
12:11	Reserved	RES	-	-
10	FILTER_BYP5 This bit enables bypassing the input noise filter for Capture Register 5, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9:8	CAPTURE_EDGE5 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 5. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
7:5	FCLK_SEL4 This 3-bit field sets the clock source for the input filter for Capture Register 4. See Table 14-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
4:3	Reserved	RES	-	-
2	FILTER_BYP4 This bit enables bypassing the input noise filter for Capture Register 4, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
1:0	CAPTURE_EDGE4 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 4. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS

14.10.4 FREE RUNNING TIMER REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	FREE_RUNNING_TIMER This register contains the current value of the Free Running Timer. A Capture Timer interrupt is signaled to the Interrupt Aggregator when this register transitions from FFFF_FFFFh to 0000_0000h. When FREE_ENABLE in the Capture and Compare Timer Control Register is ‘1’, this register is read-only. When FREE_ENABLE is ‘0’, this register may be written.	R/W	0h	RESET_SYS

14.10.5 CAPTURE 0 REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_0 This register saves the value copied from the Free Running timer on a programmed edge of ICT0.	R	0h	RESET_SYS

14.10.6 CAPTURE 1 REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_1 This register saves the value copied from the Free Running timer on a programmed edge of ICT1.	R	0h	RESET_SYS

14.10.7 CAPTURE 2 REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_2 This register saves the value copied from the Free Running timer on a programmed edge of ICT2.	R	0h	RESET_SYS

14.10.8 CAPTURE 3 REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_3 This register saves the value copied from the Free Running timer on a programmed edge of ICT3.	R	0h	RESET_SYS

14.10.9 CAPTURE 4 REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_4 This register saves the value copied from the Free Running timer on a programmed edge of ICT4.	R	0h	RESET_SYS

14.10.10 CAPTURE 5 REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_5 This register saves the value copied from the Free Running timer on a programmed edge of ICT5.	R	0h	RESET_SYS

14.10.11 COMPARE 0 REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COMPARE_0 A COMPARE 0 interrupt is generated when this register matches the value in the Free Running Timer.	R/W	0h	RESET_SYS

14.10.12 COMPARE 1 REGISTER

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COMPARE_1 A COMPARE 1 interrupt is generated when this register matches the value in the Free Running Timer.	R/W	0h	RESET_SYS

14.10.13 ICT MUX SELECT REGISTER

This register selects the pin mapping to the capture register.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:20	Mux Select for Capture 5 register.	R/W	5h	RESET_SYS
19:16	Mux Select for Capture 4 register.	R/W	4h	RESET_SYS
15:12	Mux Select for Capture 3 register.	R/W	3h	RESET_SYS
11:8	Mux Select for Capture 2 register.	R/W	2h	RESET_SYS
7:4	Mux Select for Capture 1 register.	R/W	1h	RESET_SYS
3:0	Mux Select for Capture 0 register.	R/W	0h	RESET_SYS

15.0 HIBERNATION TIMER

15.1 Introduction

The Hibernation Timer can generate a wake event to the Embedded Controller (EC) when it is in a hibernation mode. This block supports wake events up to 2 hours in duration. The timer is a 16-bit binary count-down timer that can be programmed in 30.5µs and 0.125 second increments for period ranges of 30.5µs to 2s or 0.125s to 136.5 minutes, respectively. Writing a non-zero value to this register starts the counter from that value. A wake-up interrupt is generated when the count reaches zero.

15.2 References

No references have been cited for this chapter

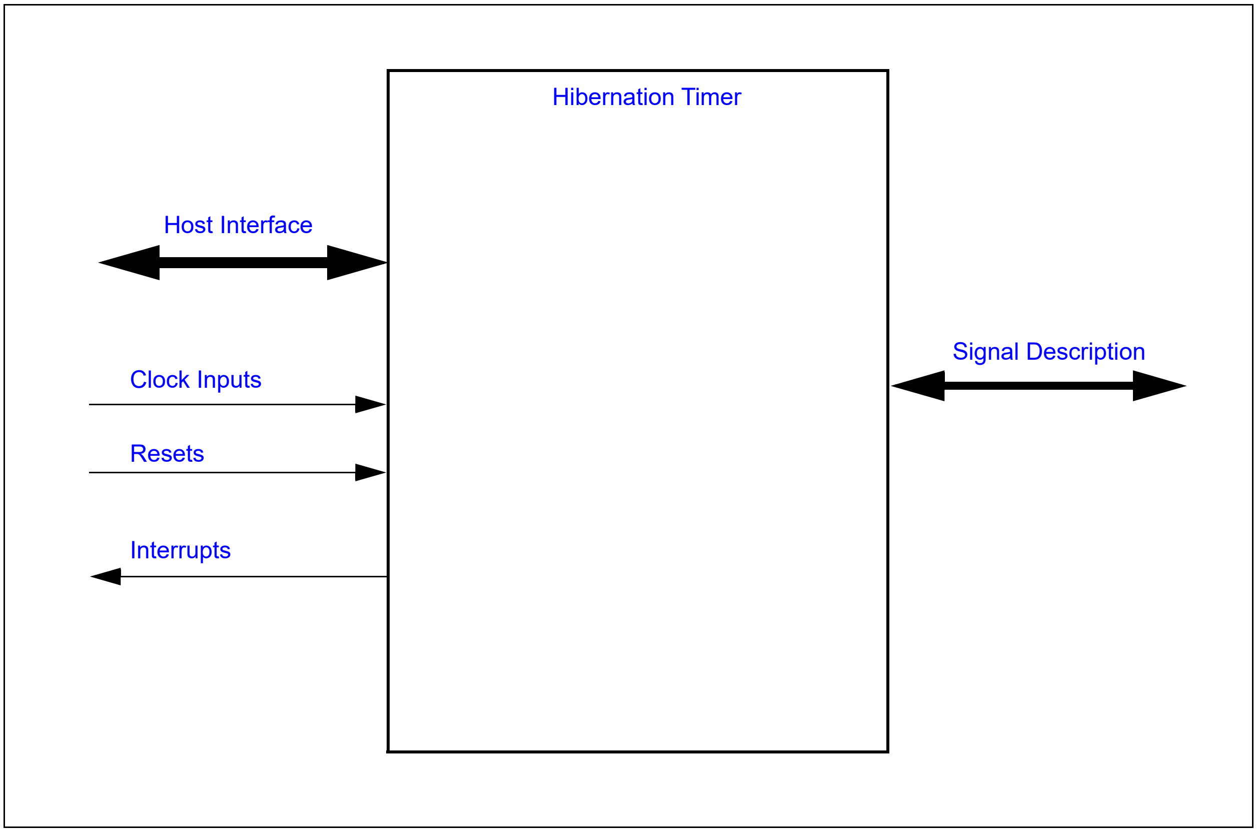
15.3 Terminology

No terms have been cited for this chapter.

15.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 15-1: HIBERNATION TIMER INTERFACE DIAGRAM



15.5 Signal Description

There are no external signals for this block.

15.6 Host Interface

The registers defined for the Hibernation Timer are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

15.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

15.7.1 POWER DOMAINS

TABLE 15-1: POWER SOURCES

Name	Description
VTR_CORE	The timer control logic and registers are all implemented on this single power domain.

15.7.2 CLOCK INPUTS

TABLE 15-2: CLOCK INPUTS

Name	Description
32KHz	This is the clock source to the timer logic. The Pre-scaler may be used to adjust the minimum resolution per bit of the counter. if the main oscillator is stopped then an external 32.768kHz clock source must be active for the Hibernation Timer to continue to operate.

15.7.3 RESETS

TABLE 15-3: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

15.8 Interrupts

This section defines the interrupt Interface signals routed to the chip interrupt aggregator.

Each instance of the Hibernation Timer in the CEC1712 can be used to generate interrupts and wake-up events when the timer decrements to zero.

TABLE 15-4: INTERRUPT INTERFACE SIGNAL DESCRIPTION TABLE

Name	Direction	Description
HTIMER	Output	Signal indicating that the timer is enabled and decrements to 0. This signal is used to generate an Hibernation Timer interrupt event.

15.9 Low Power Modes

The timer operates off of the 32KHz clock, and therefore will operate normally when the main oscillator is stopped.

The sleep enable inputs have no effect on the Hibernation Timer and the clock required outputs are only asserted during register read/write cycles for as long as necessary to propagate updates to the block core.

15.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Hibernation Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 15-5: REGISTER SUMMARY

Offset	Register Name
00h	HTimer Preload Register
04h	HTimer Control Register
08h	HTimer Count Register

15.10.1 HTIMER PRELOAD REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	HT_PRELOAD This register is used to set the Hibernation Timer Preload value. Writing this register to a non-zero value resets the down counter to start counting down from this programmed value. Writing this register to 0000h disables the hibernation counter. The resolution of this timer is determined by the CTRL bit in the HTimer Control Register. Writes to the HTimer Control Register are completed with an EC bus cycle.	R/W	000h	RESET_SYS

15.10.2 HTIMER CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	Reserved	RES	-	-
0	CTRL 1=The Hibernation Timer has a resolution of 0.125s per LSB, which yields a maximum time in excess of 2 hours. 0=The Hibernation Timer has a resolution of 30.5μs per LSB, which yields a maximum time of ~2seconds.	R	0000h	RESET_SYS

15.10.3 HTIMER COUNT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	COUNT The current state of the Hibernation Timer.	R	0000h	RESET_SYS

16.0 RTOS TIMER

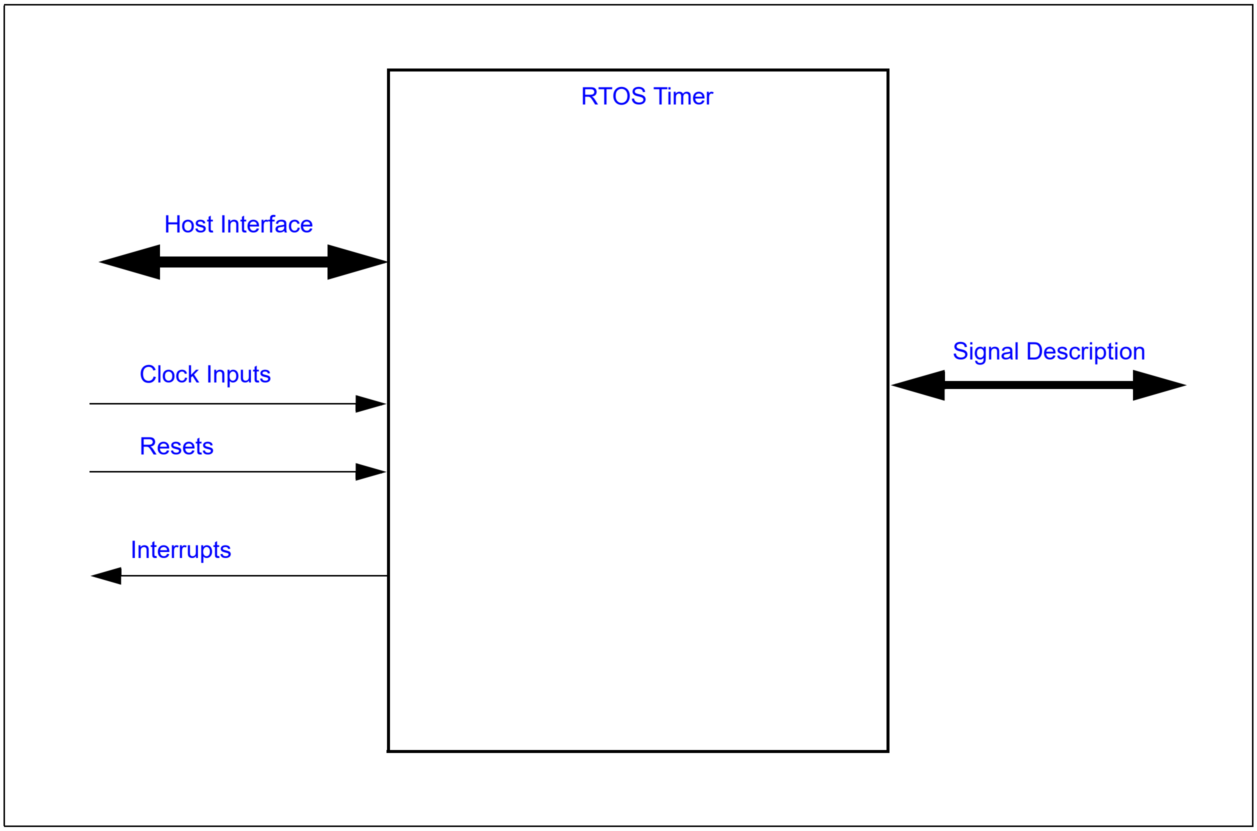
16.1 Introduction

The RTOS Timer is a low-power, 32-bit timer designed to operate on the 32kHz oscillator which is available during all chip sleep states. This allows firmware the option to sleep the processor and wake after a programmed amount of time. The timer may be used as a one-shot timer or a continuous timer. When the timer transitions to 0 it is capable of generating a wake-capable interrupt to the embedded controller. This timer may be halted during debug by hardware or via a software control bit.

16.2 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 16-1: I/O DIAGRAM OF BLOCK



16.3 Signal Description

Name	Description
HALT	RTOS Timer Halt signal. This signal is connected to the same signal that halts the embedded controller during debug (e.g., JTAG Debugger is active, break points, etc.).

16.4 Host Interface

The Embedded Controller (EC) may access this block via the registers defined in Section 16.9, "EC Registers".

16.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

16.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The timer control logic and registers are all implemented on this single power domain.

16.5.2 CLOCK INPUTS

Name	Description
32KHz	This is the clock source to the timer logic.

16.5.3 RESETS

Name	Description
RESET_SYS	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

16.6 Interrupts

Source	Description
RTOS_TIMER	RTOS Timer interrupt event. The interrupt is signaled when the timer counter transitions from 1 to 0 while counting.

16.7 Low Power Modes

The Basic Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active.

16.8 Description

The RTOS Timer is a basic down counter that can operate either as a continuous timer or a one-shot timer. When it is started, the counter is loaded with a pre-load value and counts towards 0. When the counter counts down from 1 to 0, it will generate an interrupt. In one-shot mode (the AUTO_RELOAD bit is '0'), the timer will then halt; in continuous mode (the AUTO_RELOAD bit is '1'), the counter will automatically be restarted with the pre-load value.

The timer counter can be halted by firmware by setting the FIRMWARE_TIMER_HALT bit to '1'. In addition, if enabled, the timer counter can be halted by the external HALT signal.

16.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the RTOS Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 16-1: REGISTER SUMMARY

Offset	Register Name
00h	RTOS Timer Count Register
04h	RTOS Timer Preload Register
08h	RTOS Timer Control Register
0Ch	Soft Interrupt Register

16.9.1 RTOS TIMER COUNT REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER This register contains the current value of the RTOS Timer counter. This register should be read as a DWORD. There is no latching mechanism of the upper bytes implemented if the register is accessed as a byte or word. Reading the register with byte or word operations may give incorrect results.	R/W	0h	RESET_SYS

16.9.2 RTOS TIMER PRELOAD REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PRE_LOAD The this register is loaded into the RTOS Timer counter either when the TIMER_START bit is written with a ‘1’, or when the timer counter counts down to ‘0’ and the AUTO_RELOAD bit is ‘1’. This register must be programmed with a new count value before the TIMER_START bit is set to ‘1’. If this register is updated while the counter is operating, the new count value will only take effect if the counter transitions form 1 to 0 while the AUTO_RELOAD bit is set.	R/W	0h	RESET_SYS

16.9.3 RTOS TIMER CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	RES	-	-
4	FIRMWARE_TIMER_HALT 1=The timer counter is halted. If the counter was running, clearing this bit will restart the counter from the value at which it halted 0=The timer counter, if enabled, will continue to run	R/W	0h	RESET_SYS
3	EXT_HARDWARE_HALT_EN 1=The timer counter is halted when the external HALT signal is asserted. Counting is always enabled if HALT is de-asserted. 0=The HALT signal does not affect the RTOS Timer	R/W	0h	RESET_SYS
2	TIMER_START Writing a ‘1’ to this bit will load the timer counter with the RTOS Timer Preload Register and start counting. If the Preload Register is 0, counting will not start and this bit will be cleared to ‘0’. Writing a ‘0’ to this bit will halt the counter and clear its contents to 0. The RTOS timer interrupt will not be generated. This bit is automatically cleared if the AUTO_RELOAD bit is ‘0’ and the timer counter transitions from 1 to 0.	R/W	0h	RESET_SYS
1	AUTO_RELOAD 1=The the RTOS Timer Preload Register is loaded into the timer counter and the counter is restarted when the counter transitions from 1 to 0 0=The timer counter halts when it transitions from 1 to 0 and will not restart	R/W	0h	RESET_SYS
0	BLOCK_ENABLE 1=RTOS timer counter is enabled 0=RTOS timer disabled. All register bits are reset to their default state	R/W	0h	RESET_SYS

16.9.4 SOFT INTERRUPT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	SWI_3 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS
2	SWI_2 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS
1	SWI_1 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS
0	SWI_0 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS

17.0 REAL TIME CLOCK

17.1 Introduction

This block provides the capabilities of an industry-standard 146818B Real-Time Clock module, without CMOS RAM. Enhancements to this architecture include:

- Industry standard Day of Month Alarm field, allowing for monthly alarms
- Configurable, automatic Daylight Savings adjustment
- Week Alarm for periodic interrupts and wakes based on Day of Week
- System Wake capability on interrupts.

17.2 References

1. Motorola 146818B Data Sheet, available on-line
2. Intel Lynx Point PCH EDS specification

17.3 Terminology

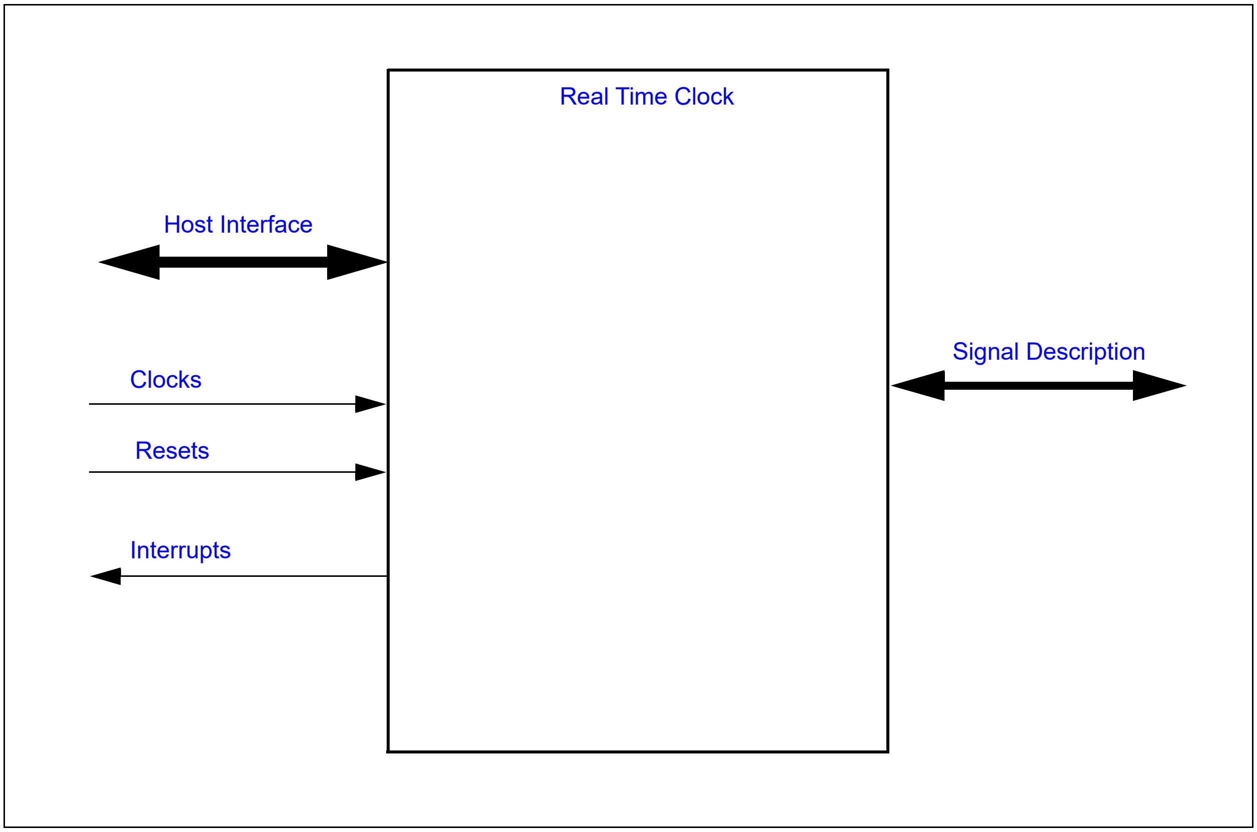
Time and Date Registers:

This is the set of registers that are automatically counted by hardware every 1 second while the block is enabled to run and to update. These registers are: **Seconds**, **Minutes**, **Hours**, **Day of Week**, **Day of Month**, **Month**, and **Year**.

17.4 Interface

This block's connections are entirely internal to the chip.

FIGURE 17-1: I/O DIAGRAM OF BLOCK



17.5 Signal Description

There are no external signals.

17.6 Host Interface

The registers defined for the Real Time Clock are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

17.7 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

17.7.1 POWER DOMAINS

TABLE 17-1: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR_CORE	This power well sources only host register accesses. The block continues to operate internally while this rail is down.

17.7.2 CLOCKS

TABLE 17-2: CLOCKS

Name	Description
32KHz	This clock input drives all internal logic, and will be present at all times that the VBAT well is powered.

17.7.3 RESETS

TABLE 17-3: RESET SIGNALS

Name	Description
RESET_VBAT	This reset signal is used in the RESET_RTC signal to reset all of the registers and logic in this block. It directly resets the Soft Reset bit in the RTC Control Register.
RESET_RTC	This reset signal resets all of the registers and logic in this block, except for the Soft Reset bit in the RTC Control Register. It is triggered by RESET_VBAT, but can also be triggered by a SOFT_RESET from the RTC Control Register.
RESET_SYS	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR_CORE powered circuitry on-chip. Otherwise it has no effect on the internal state.
SOFT_RESET	This is the block reset and resets all the registers and logic in the block

17.8 Interrupts

TABLE 17-4: SYSTEM INTERRUPTS

Source	Description
RTC	This interrupt source for the SIRQ logic is generated when any of the following events occur: • Update complete. This is triggered, at 1-second intervals, when the Time register updates have completed • Alarm. This is triggered when the alarm value matches the current time (and date, if used) • Periodic. This is triggered at the chosen programmable rate

TABLE 17-5: EC INTERRUPTS

Source	Description
RTC	This interrupt is signaled to the Interrupt Aggregator when any of the following events occur: • Update complete. This is triggered, at 1-second intervals, when the Time register updates have completed • Alarm. This is triggered when the alarm value matches the current time (and date, if used) • Periodic. This is triggered at the chosen programmable rate
RTC ALARM	This wake interrupt is signaled to the Interrupt Aggregator when an Alarm event occurs.

17.9 Low Power Modes

The RTC has no low-power modes. It runs continuously while the VBAT well is powered.

17.10 Description

This block provides the capabilities of an industry-standard 146818B Real-Time Clock module, excluding the CMOS RAM and the SQW output. See the following registers, which represent enhancements to this architecture. These enhancements are listed below.

See the Date Alarm field of Register D for a Day of Month qualifier for alarms.

See the Week Alarm Register for a Day of Week qualifier for alarms.

See the registers Daylight Savings Forward Register and Daylight Savings Backward Register for setting up hands-off Daylight Savings adjustments.

See the RTC Control Register for enhanced control over the block’s operations.

17.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Real Time Clock. Host access for each register listed in this table is defined as an offset in the Host address space to the Block’s Base Address, as defined by the instance’s Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the Real Time Clock shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “Offset” column.

TABLE 17-6: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Seconds Register
01h	Seconds Alarm Register
02h	Minutes Register
03h	Minutes Alarm Register
04h	Hours Register
05h	Hours Alarm Register
06h	Day of Week Register
07h	Day of Month Register
08h	Month Register
09h	Year Register
0Ah	Register A
0Bh	Register B

TABLE 17-6: RUNTIME REGISTER SUMMARY (CONTINUED)

Offset	Register Name
0Ch	Register C
0Dh	Register D
0Eh	Reserved
0Fh	Reserved
10h	RTC Control Register
14h	Week Alarm Register
18h	Daylight Savings Forward Register
1Ch	Daylight Savings Backward Register
20h	TEST

Note: This extended register set occupies offsets that have historically been used as CMOS RAM. Code ported to use this block should be examined to ensure that it does not assume that RAM exists in this block.

17.11.1 SECONDS REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SECONDS Displays the number of seconds past the current minute, in the range 0--59. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

17.11.2 SECONDS ALARM REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SECONDS_ALARM Holds a match value, compared against the Seconds Register to trigger the Alarm event. Values written to this register must use the format defined by the current setting of the DM bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RESET_RTC

17.11.3 MINUTES REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MINUTES Displays the number of minutes past the current hour, in the range 0--59. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

17.11.4 MINUTES ALARM REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MINUTES_ALARM Holds a match value, compared against the Minutes Register to trigger the Alarm event. Values written to this register must use the format defined by the current setting of the DM bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RESET_RTC

17.11.5 HOURS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	HOURS_AM_PM In 12-hour mode (see bit "24/12" in register B), this bit indicates AM or PM. 1=PM 0=AM	R/W	0b	RESET_RTC
6:0	HOURS Displays the number of the hour, in the range 1--12 for 12-hour mode (see bit "24/12" in register B), or in the range 0--23 for 24-hour mode. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

17.11.6 HOURS ALARM REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOURS_ALARM Holds a match value, compared against the Hours Register to trigger the Alarm event. Values written to this register must use the format defined by the current settings of the DM bit and the 24/12 bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RESET_RTC

17.11.7 DAY OF WEEK REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DAY_OF_WEEK Displays the day of the week, in the range 1 (Sunday) through 7 (Saturday). Numbers in this range are identical in both binary and BCD notation, so this register's format is unaffected by the DM bit.	R/W	00h	RESET_RTC

17.11.8 DAY OF MONTH REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DAY_OF_MONTH Displays the day of the current month, in the range 1--31. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

17.11.9 MONTH REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MONTH Displays the month, in the range 1--12. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

17.11.10 YEAR REGISTER

Offset	09h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	YEAR Displays the number of the year in the current century, in the range 0 (year 2000) through 99 (year 2099). Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

17.11.11 REGISTER A

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UPDATE_IN_PROGRESS ‘0’ indicates that the Time and Date registers are stable and will not be altered by hardware soon. ‘1’ indicates that a hardware update of the Time and Date registers may be in progress, and those registers should not be accessed by the host program. This bit is set to ‘1’ at a point 488us (16 cycles of the 32K clock) before the update occurs, and is cleared immediately after the update. See also the Update-Ended Interrupt, which provides more useful status.	R	0b	RESET_RTC
6:4	DIVISION_CHAIN_SELECT This field provides general control for the Time and Date register updating logic. 11xb=Halt counting. The next time that 010b is written, updates will begin 500ms later. 010b=Required setting for normal operation. It is also necessary to set the Block Enable bit in the RTC Control Register to ‘1’ for counting to begin 000b=Reserved. This field should be initialized to another value before Enabling the block in the RTC Control Register Other values Reserved	R/W	000b	RESET_RTC
3:0	RATE_SELECT This field selects the rate of the Periodic Interrupt source. See Table 17-7	R/W	0h	RESET_RTC

TABLE 17-7: REGISTER A FIELD RS: PERIODIC INTERRUPT SETTINGS

RS (hex)	Interrupt Period
0	Never Triggered
1	3.90625 ms
2	7.8125 ms
3	122.070 us
4	244.141 us
5	488.281 us
6	976.5625 us
7	1.953125 ms
8	3.90625 ms
9	7.8125 ms
A	15.625 ms
B	31.25 ms
C	62.5 ms
D	125 ms
E	250 ms
F	500 ms

17.11.12 REGISTER B

Offset	0Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UPDATE_CYCLE_INHIBIT In its default state '0', this bit allows hardware updates to the Time and Date registers, which occur at 1-second intervals. A '1' written to this field inhibits updates, allowing these registers to be cleanly written to different values. Writing '0' to this bit allows updates to continue.	R/W	0b	RESET_RTC
6	PERIODIC_INTERRUPT_ENABLE 1=Alows the Periodic Interrupt events to be propagated as interrupts 0=Periodic events are not propagates as interrupts	R/W	0b	RESET_RTC
5	ALARM_INTERRUPT_ENABLE 1=Alows the Alarm Interrupt events to be propagated as interrupts 0=Alarm events are not propagates as interrupts	R/W	0b	RESET_RTC
4	UPDATE_ENDED_INTERRUPT_ENABLE 1=Alows the Update Ended Interrupt events to be propagated as interrupts 0=Update Ended events are not propagates as interrupts	R/W	0b	RESET_RTC
3	Reserved	RES	-	-
2	DATA_MODE 1=Binary Mode for Dates and Times 0=BCD Mode for Dates and Times	R/W	0b	RESET_RTC
1	HOUR_FORMAT_24_12 1=24-Hour Format for Hours and Hours Alarm registers. 24-Hour format keeps the AM/PM bit off, with value range 0--23 0=12-Hour Format for Hours and Hours Alarm registers. 12-Hour format has an AM/PM bit, and value range 1--12	R/W	0b	RESET_RTC
0	DAYLIGHT_SAVINGS_ENABLE 1=Enables automatic hardware updating of the hour, using the registers Daylight Savings Forward and Daylight Savings Backward to select the yearly date and hour for each update 0=Automatic Daylight Savings updates disabled	R/W	0b	RESET_RTC

Note: The DATA_MODE and HOUR_FORMAT_24_12 bits affect only how values are presented as they are being read and how they are interpreted as they are being written. They do not affect the internal contents or interpretations of registers that have already been written, nor do they affect how those registers are represented or counted internally. This mode bits may be set and cleared dynamically, for whatever I/O data representation is desired by the host program.

17.11.13 REGISTER C

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	INTERRUPT_REQUEST_FLAG 1=Any of bits[6:4] below is active after masking by their respective Enable bits in Register B. 0=No bits in this register are active This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
6	PERIODIC_INTERRUPT_FLAG 1=A Periodic Interrupt event has occurred since the last time this register was read. This bit displays status regardless of the Periodic Interrupt Enable bit in Register B 0=A Periodic Interrupt event has not occurred This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
5	ALARM_FLAG 1=An Alarm event has occurred since the last time this register was read. This bit displays status regardless of the Alarm Interrupt Enable bit in Register B. 0=An Alarm event has not occurred This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
4	UPDATE_ENDED_INTERRUPT_FLAG 1=A Time and Date update has completed since the last time this register was read. This bit displays status regardless of the Update-Ended Interrupt Enable bit in Register B. Presentation of this status indicates that the Time and Date registers will be valid and stable for over 999ms 0=A Time and Data update has not completed since the last time this register was read This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
3:0	Reserved	RES	-	-

17.11.14 REGISTER D

Offset	0Dh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	RES	-	-
5:0	DATE_ALARM This field, if set to a non-zero value, will inhibit the Alarm interrupt unless this field matches the contents of the Month register also. If this field contains 00h (default), it represents a don't-care, allowing more frequent alarms.	R/W	00h	RESET_RTC

17.11.15 RTC CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	Reserved	RES	-	-
3	ALARM_ENABLE 1=Enables the Alarm features 0=Disables the Alarm features	R/W	0b	RESET_RTC
2	VCI_ENABLE 1= RTC Alarm event is routed to chip level VCI Circuitry 0= RTC alarm event is inhibited from affecting the VCI circuitry	R/W	0b	RESET_RTC
1	SOFT_RESET A ‘1’ written to this bit position will trigger the RESET_RTC reset, resetting the block and all registers except this one and the Test Register. This bit is self-clearing at the end of the reset.	R/W	0b	RESET_VBAT
0	BLOCK_ENABLE This bit must be ‘1’ in order for the block to function internally. Registers may be initialized first, before setting this bit to ‘1’ to start operation.	R/W	0b	RESET_RTC

17.11.16 WEEK ALARM REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ALARM_DAY_OF_WEEK This register, if written to a value in the range 1--7, will inhibit the Alarm interrupt unless this field matches the contents of the Day of Week Register also. If this field is written to any value 11xxxxxxb (like the default FFh), it represents a don’t-care, allowing more frequent alarms, and will read back as FFh until another value is written.	R/W	FFh	RESET_RTC

17.11.17 DAYLIGHT SAVINGS FORWARD REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	DST_FORWARD_AM_PM This bit selects AM vs. PM, to match bit[7] of the Hours Register if 12-Hour mode is selected in Register B at the time of writing.	R/W	0b	RESET_RTC
30:24	DST_FORWARD_HOUR This field holds the matching value for bits[6:0] of the Hours register. The written value will be interpreted according to the 24/12 Hour mode and DM mode settings at the time of writing.	R/W	00h	RESET_RTC
23:19	Reserved	RES	-	-

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18:16	DST_FORWARD_WEEK This value matches an internally-maintained week number within the current month. Valid values for this field are: 5=Last week of month 4 =Fourth week of month 3=Third week of month 2=Second week of month 1=First week of month	R/W	0h	RESET _RTC
15:11	Reserved	RES	-	-
10:8	DST_FORWARD_DAY_OF_WEEK This field matches the Day of Week Register bits[2:0].	R/W	0h	RESET _RTC
7:0	DST_FORWARD_MONTH This field matches the Month Register.	R/W	00h	RESET _RTC

This is a 32-bit register, accessible also as individual bytes. When writing as individual bytes, ensure that the DSE bit (in Register B) is off first, or that the block is disabled or stopped (SET bit), to prevent a time update while this register may have incompletely-updated contents.

When enabled by the DSE bit in Register B, this register defines an hour and day of the year at which the Hours register will be automatically incremented by 1 additional hour.

There are no don't-care fields recognized. All fields must be already initialized to valid settings whenever the DSE bit is '1'.

Fields other than Week and Day of Week use the current setting of the DM bit (binary vs. BCD) to interpret the information as it is written to them. Their values, as held internally, are not changed by later changes to the DM bit, without subsequently writing to this register as well.

Note: An Alarm that is set inside the hour after the time specified in this register will not be triggered, because that one-hour period is skipped. This period includes the exact time (0 minutes: 0 seconds) given by this register, through the 59 minutes: 59 seconds point afterward.

17.11.18 DAYLIGHT SAVINGS BACKWARD REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	DST_BACKWARD_AM_PM This bit selects AM vs. PM, to match bit[7] of the Hours register if 12-Hour mode is selected in Register B at the time of writing.	R/W	0b	RESET _RTC
30:24	DST_BACKWARD_HOUR This field holds the matching value for bits[6:0] of the Hours register. The written value will be interpreted according to the 24/12 Hour mode and DM mode settings at the time of writing.	R/W	00h	RESET _RTC
23:19	Reserved	RES	-	-

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18:16	DST_BACKWARD_WEEK This value matches an internally-maintained week number within the current month. Valid values for this field are: 5=Last week of month 4 =Fourth week of month 3=Third week of month 2=Second week of month 1=First week of month	R/W	0h	RESET_RTC
15:11	Reserved	RES	-	-
10:8	DST_BACKWARD_DAY_OF_WEEK This field matches the Day of Week Register bits[2:0].	R/W	0h	RESET_RTC
7:0	DST_BACKWARD_MONTH This field matches the Month Register.	R/W	00h	RESET_RTC

This is a 32-bit register, accessible also as individual bytes. When writing as individual bytes, ensure that the DSE bit (in Register B) is off first, or that the block is disabled or stopped (SET bit), to prevent a time update while this register may have incompletely-updated contents.

When enabled by the DSE bit in Register B, this register defines an hour and day of the year at which the Hours register increment will be inhibited from occurring. After triggering, this feature is automatically disabled for long enough to ensure that it will not retrigger the second time this Hours value appears, and then this feature is re-enabled automatically.

There are no don’t-care fields recognized. All fields must be already initialized to valid settings whenever the DSE bit is ‘1’.

Fields other than Week and Day of Week use the current setting of the DM bit (binary vs. BCD) to interpret the information as it is written to them. Their values, as held internally, are not changed by later changes to the DM bit, without subsequently writing to this register as well.

Note: An Alarm that is set inside the hour before the time specified in this register will be triggered twice, because that one-hour period is repeated. This period will include the exact time (0 minutes: 0 seconds) given by this register, through the 59 minutes: 59 seconds point afterward.

18.0 WEEK TIMER

18.1 Introduction

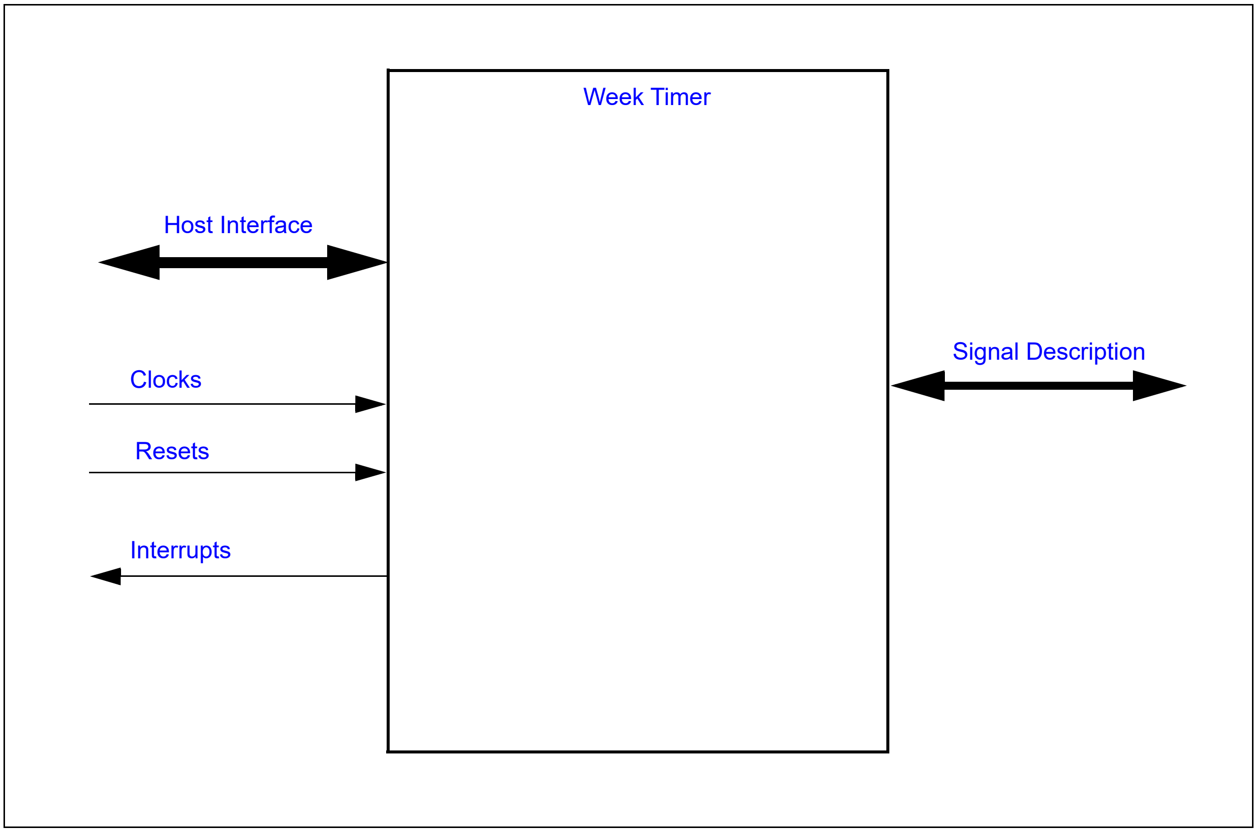
The Week Alarm Interface provides two timekeeping functions: a Week Timer and a Sub-Week Timer. Both the Week Timer and the Sub-Week Timer assert the Power-Up Event Output which automatically powers-up the system from the G3 state. Features include:

- EC interrupts based on matching a counter value
- Repeating interrupts at 1 second and sub-1 second intervals
- System Wake capability on interrupts, including Wake from Heavy Sleep

18.2 Interface

This block's connections are entirely internal to the chip.

FIGURE 18-1: I/O DIAGRAM OF BLOCK



18.3 Signal Description

TABLE 18-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
BGPO	OUTPUT	Battery-powered general purpose outputs
SYSPWR_PRES	INPUT	Input signal used to gate the POWER_UP_EVENT

Note 1: Please refer to TABLE 1-1: for the number of BGPO's and SYSPWR_PRES availability in the package.

18.4 Host Interface

The registers defined for the Week Timer are accessible only by the EC.

18.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

18.5.1 POWER DOMAINS

TABLE 18-3: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR_CORE	This power well sources only host register accesses. The block continues to operate internally while this rail is down.

18.5.2 CLOCKS

TABLE 18-4: CLOCKS

Name	Description
48MHz	Clock used for host register access
32KHz	This 32KHz clock input drives all internal logic, and will be present at all times that the VBAT well is powered.

18.5.3 RESETS

TABLE 18-5: RESET SIGNALS

Name	Description
RESET_VBAT	This reset signal is used reset all of the registers and logic in this block.
RESET_SYS	This reset signal is used to inhibit the Host register access and isolates this block from VTR_CORE powered circuitry on-chip. Otherwise it has no effect on the internal state.

18.6 Interrupts

TABLE 18-6: EC INTERRUPTS

Source	Description
WEEK_ALARM_INT	This interrupt is signaled to the Interrupt Aggregator when the Week Alarm Counter Register is greater than or equal to the Week Timer Compare Register. The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SUB_WEEK_ALARM_INT	This interrupt is signaled to the Interrupt Aggregator when the Sub-Week Alarm Counter Register decrements from ‘1’ to ‘0’. The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
ONE_SECOND	This interrupt is signaled to the Interrupt Aggregator at an isochronous rate of once per second. The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SUB_SECOND	This interrupt is signaled to the Interrupt Aggregator at an isochronous rate programmable between 0.5Hz and 32.768KHz. The rate interrupts are signaled is determined by the SPISR field in the Sub-Second Programmable Interrupt Select Register. See Table 18-9, "SPISR Encoding". The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.

18.7 Low Power Modes

The Week Alarm has no low-power modes. It runs continuously while the VBAT well is powered.

18.8 Power-Up Events

The Week Timer POWER_UP_EVENT can be used to power up the system after a timed interval.

The POWER_UP_EVENT can be asserted under the following two conditions:

1. The Week Alarm Counter Register is greater than or equal to the Week Timer Compare Register
2. The Sub-Week Alarm Counter Register decrements from ‘1’ to ‘0’

The assertion of the POWER_UP_EVENT is inhibited if the POWERUP_EN field in the Control Register is ‘0’

Once a POWER_UP_EVENT is asserted the POWERUP_EN bit must be cleared to reset the output. Clearing POWERUP_EN is necessary to avoid unintended power-up cycles.

18.9 Description

The Week Alarm block provides battery-powered timekeeping functions, derived from a low-power 32KHz clock, that operate even when the device’s main power is off. The block contains a set of counters that can be used to generate one-shot and periodic interrupts to the EC for periods ranging from about 30 microseconds to over 8 years. The Week Alarm can be used in conjunction with the VBAT-Powered Control Interface to power up a sleeping system after a configurable period.

In addition to basic timekeeping, the Week Alarm block can be used to control the battery-powered general purpose BGPO outputs.

18.9.1 INTERNAL COUNTERS

The Week Timer includes 3 counters:

18.9.1.1 28-bit Week Alarm Counter

This counter is 28 bits wide. The clock for this counter is the overflow of the Clock Divider, and as long as the Week Timer is enabled, it is incremented at a 1 Hz rate.

Both an interrupt and a power-up event can be generated when the contents of this counter matches the contents of the Week Timer Compare Register.

18.9.1.2 9-bit Sub-Week Alarm Counter

This counter is 9 bits wide. It is decremented by 1 at each tick of its selected clock. It can be configured either as a one-shot or repeating event generator.

Both an interrupt and a power-up event can be generated when this counter decrements from 1 to 0.

The Sub-Week Alarm Counter can be configured with a number of different clock sources for its time base, derived from either the Week Alarm Counter or the Clock Divider, by setting the SUBWEEK_TICK field of the Sub-Week Control Register.

TABLE 18-7: SUB-WEEK ALARM COUNTER CLOCK

SUBWEEK_ TICK	Source	SPISR	Frequency	Minimum Duration	Maximum Duration
0	Counter Disabled				
1	Sub-Second	0	Counter Disabled		
		1	2 Hz	500 ms	255.5 sec
		2	4 Hz	250 ms	127.8 sec
		3	8 Hz	125 ms	63.9 sec
		4	16 Hz	62.5	31.9 sec
		5	32 Hz	31.25 ms	16.0 sec
		6	64 Hz	15.6 ms	8 sec
		7	128 Hz	7.8 ms	4 sec
		8	256 Hz	3.9 ms	2 sec
		9	512 Hz	1.95 ms	1 sec
		10	1024 Hz	977 µS	499 ms
		11	2048 Hz	488 µS	249.5 ms
		12	4096 Hz	244 µS	124.8 ms
		13	8192 Hz	122 µS	62.4 ms
		14	16.384 KHz	61.1 µS	31.2 ms
		15	32.768 KHz	30.5 µS	15.6 ms
2	Second	n/a	1 Hz	1 sec	511 sec
3	Reserved				
4	Week Counter bit 3	n/a	125 Hz	8 sec	68.1 min
5	Week Counter bit 5	n/a	31.25 Hz	32 sec	272.5 min
6	Week Counter bit 7	n/a	7.8125 Hz	128 sec	18.17 hour
7	Week Counter bit 9	n/a	1.95 Hz	512 sec	72.68 hour

Note 1: The Week Alarm Counter **must not** be modified by firmware if Sub-Week Alarm Counter is using the Week Alarm Counter as its clock source (i.e., the SUBWEEK_TICK field is set to any of the values 4, 5, 6 or 7). The Sub-Week Alarm Counter must be disabled before changing the Week Alarm Counter. For example, the following sequence may be used:

1.Write 0h to the Sub-Week Alarm Counter Register (disabling the Sub-Week Counter)

2.Write the Week Alarm Counter Register

3.Write a new value to the Sub-Week Alarm Counter Register, restarting the Sub-Week Counter

18.9.1.3 15-bit Clock Divider

This counter is 15 bits wide. The clock for this counter is 32KHz, and as long as the Week Timer is enabled, it is incremented at 32.768KHz rate. The Clock Divider automatically The Clock Divider generates a clock out of 1 Hz when the counter wraps from 7FFFh to 0h.

By selecting one of the 15 bits of the counter, using the Sub-Second Programmable Interrupt Select Register, the Clock Divider can be used either to generate a time base for the Sub-Week Alarm Counter or as an isochronous interrupt to the EC, the SUB_SECOND interrupt. See Table 18-9, "SPISR Encoding" for a list of available frequencies.

18.9.2 TIMER VALID STATUS

If power on reset occurs on the VBAT power rail while the main device power is off, the counters in the Week Alarm are invalid. If firmware detects a POR on the VBAT power rail after a system boot, by checking the status bits in the Power, Clocks and Resets registers, the Week Alarm block must be reinitialized.

18.9.3 APPLICATION NOTE: REGISTER TIMING

Register writes in the Week Alarm complete within two cycles of the 32KHz clock.The write completes even if the main system clock is stopped before the two cycles of the 32K clock complete. Register reads complete in one cycle of the internal bus clock.

All Week Alarm interrupts that are asserted within the same cycle of the 32KHz clock are synchronously asserted to the EC.

18.9.4 APPLICATION NOTE: USE OF THE WEEK TIMER AS A 43-BIT COUNTER

The Week Timer cannot be directly used as a 42-bit counter that is incremented directly by the 32.768KHz clock domain. The upper 28 bits (28-bit Week Alarm Counter) are incremented at a 1Hz rate and the lower 16 bits (15-bit Clock Divider) are incremented at a 32.768KHz rate, but the increments are not performed in parallel. In particular, the upper 28 bits are incremented when the lower 15 bits increment from 0 to 1, so as long as the Clock Divider Register is 0 the two registers together, treated as a single value, have a smaller value then before the lower register rolled over from 7FFFh to 0h.

The following code can be used to treat the two registers as a single large counter. This example extracts a 32-bit value from the middle of the 43-bit counter:

```
dword TIME_STAMP(void)
{
    AHB_dword wct_value;
    AHB_dword cd_value1;
    AHB_dword cd_value2;
    dword irqEnableSave;

    //Disable interrupts
    irqEnableSave = IRQ_ENABLE;
    IRQ_ENABLE = 0;

    //Read 15-bit clk divider reading register, save result in A
    cd_value1 = WTIMER->CLOCK_DIVIDER;
    //Read 28 bit up-counter timer register, save result in B
    wct_value = WTIMER->WEEK_COUNTER_TIMER;
    //Read 15-bit clk divider reading register, save result in C
    cd_value2 = WTIMER->CLOCK_DIVIDER;

    if (0 == cd_value2)
    {
        wct_value = wct_value + 1;
    }
    else if ( (cd_value2 < cd_value1) || (0 == cd_value1))
    {
        wct_value = WTIMER->WEEK_COUNTER_TIMER;
    }

    //Enable interrupts
    IRQ_ENABLE = irqEnableSave;

    return (WTIMER_BASE + ((wct_value << 10) | (cd_value2>>5)));
}
```

18.10 Battery-Powered General Purpose Outputs

The Week Timer contains the control logic for Battery-Powered General Purposes Outputs (BGPOs). These are output-only pins whose state can be controlled by firmware and preserved when the device is operating on VBAT power alone.

18.11 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Week Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 18-8: REGISTER SUMMARY

Offset	Register Name
00h	Control Register
04h	Week Alarm Counter Register
08h	Week Timer Compare Register
0Ch	Clock Divider Register
10h	Sub-Second Programmable Interrupt Select Register
14h	Sub-Week Control Register
18h	Sub-Week Alarm Counter Register

18.11.1 CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	POWERUP_EN This bit controls the state of the Power-Up Event Output and enables Week POWER-UP Event decoding in the VBAT-Powered Control Interface. See Section 18.8, "Power-Up Events" for a functional description of the POWER-UP_EN bit. 1=Power-Up Event Output Enabled 0=Power-Up Event Output Disabled and Reset	R/W	00h	RESET_VBAT
5:1	Reserved	RES	-	-
0	WT_ENABLE The WT_ENABLE bit is used to start and stop the Week Alarm Counter Register and the Clock Divider Register. The value in the Counter Register is held when the WT_ENABLE bit is not asserted ('0') and the count is resumed from the last value when the bit is asserted ('1'). The 15-Bit Clock Divider is reset to 00h and the Week Alarm Interface is in its lowest power consumption state when the WT_ENABLE bit is not asserted.	R/W	1h	RESET_VBAT

18.11.2 WEEK ALARM COUNTER REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	RES	-	-
27:0	WEEK_COUNTER While the WT_ENABLE bit is '1', this register is incremented at a 1 Hz rate. Writes of this register may require one second to take effect. Reads return the current state of the register. Reads and writes complete independently of the state of WT_ENABLE.	R/W	00h	RESET_VBAT

18.11.3 WEEK TIMER COMPARE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	RES	-	-
27:0	WEEK_COMPARE A Week Alarm Interrupt and a Week Alarm Power-Up Event are asserted when the Week Alarm Counter Register is greater than or equal to the contents of this register. Reads and writes complete independently of the state of WT_ENABLE.	R/W	FFFFFFFh	RESET_VBAT

18.11.4 CLOCK DIVIDER REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:0	CLOCK_DIVIDER Reads of this register return the current state of the Week Timer 15-bit clock divider.	R	-	RESET_VBAT

18.11.5 SUB-SECOND PROGRAMMABLE INTERRUPT SELECT REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	SPISR This field determines the rate at which Sub-Second interrupt events are generated. Table 18-9, "SPISR Encoding" shows the relation between the SPISR encoding and Sub-Second interrupt rate.	R/W	00h	RESET_VBAT

TABLE 18-9: SPISR ENCODING

SPISR Value	Sub-Second Interrupt Rate, Hz	Interrupt Period
0	Interrupts disabled	
1	2	500 ms
2	4	250 ms
3	8	125 ms

TABLE 18-9: SPISR ENCODING (CONTINUED)

SPISR Value	Sub-Second Interrupt Rate, Hz	Interrupt Period
4	16	62.5 ms
5	32	31.25 ms
6	64	15.63 ms
7	128	7.813 ms
8	256	3.906 ms
9	512	1.953 ms
10	1024	977 µS
11	2048	488 µS
12	4096	244 µS
13	8192	122 µS
14	16384	61 µS
15	32768	30.5 µS

18.11.6 SUB-WEEK CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	RES	-	-
9:7	SUBWEEK_TICK This field selects the clock source for the Sub-Week Counter. See Table 18-7, "Sub-Week Alarm Counter Clock" for the description of the options for this field. See also Note 1.	R/W	0	RESET _VBAT
6	AUTO_RELOAD 1= No reload occurs when the Sub-Week Counter expires 0= Reloads the SUBWEEK_COUNTER_LOAD field into the Sub-Week Counter when the counter expires.	R/W	0	RESET _VBAT
5	SYSPWR_PRES_ENABLE This bit controls whether the SYSPWR_PRES input pin has an effect on the POWER_UP_EVENT signal from this block. 1=The POWER_UP_EVENT will only be asserted if the SYSPWR_PRES input is high. If the SYSPWR_PRES input is low, the POWER_UP_EVENT will not be asserted 0=The SYSPWR_PRES input is ignored. It has no effect on the POWER_UP_EVENT	R/W	0	RESET _VBAT
4	SYSPWR_PRES_STATUS This bit provides the current state of the SYSPWR_PRES input pin.	R	-	RESET _VBAT
5	TEST Must always be written with 0.	R/W	0	-
4:2	Reserved	RES	-	-

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	WEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Week Alarm Counter Register is greater than or equal the contents of the Week Timer Compare Register and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit does not have to be cleared to remove a Week Timer Power-Up Event.	R/WC	0	RESET_VBAT
0	SUBWEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Sub-Week Alarm Counter Register decrements from '1' to '0' and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit MUST be cleared to remove a Sub-Week Timer Power-Up Event.	R/WC	0	RESET_VBAT

18.11.7 SUB-WEEK ALARM COUNTER REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	Reserved	RES	-	-
24:16	SUBWEEK_COUNTER_STATUS Reads of this register return the current state of the 9-bit Sub-Week Alarm counter.	R	00h	RESET_VBAT
15:9	Reserved	RES	-	-
8:0	SUBWEEK_COUNTER_LOAD Writes with a non-zero value to this field reload the 9-bit Sub-Week Alarm counter. Writes of 0 disable the counter. If the Sub-Week Alarm counter decrements to 0 and the AUTO_RELOAD bit is set, the value in this field is automatically loaded into the Sub-Week Alarm counter.	R/W	00h	RESET_VBAT

19.0 TACH

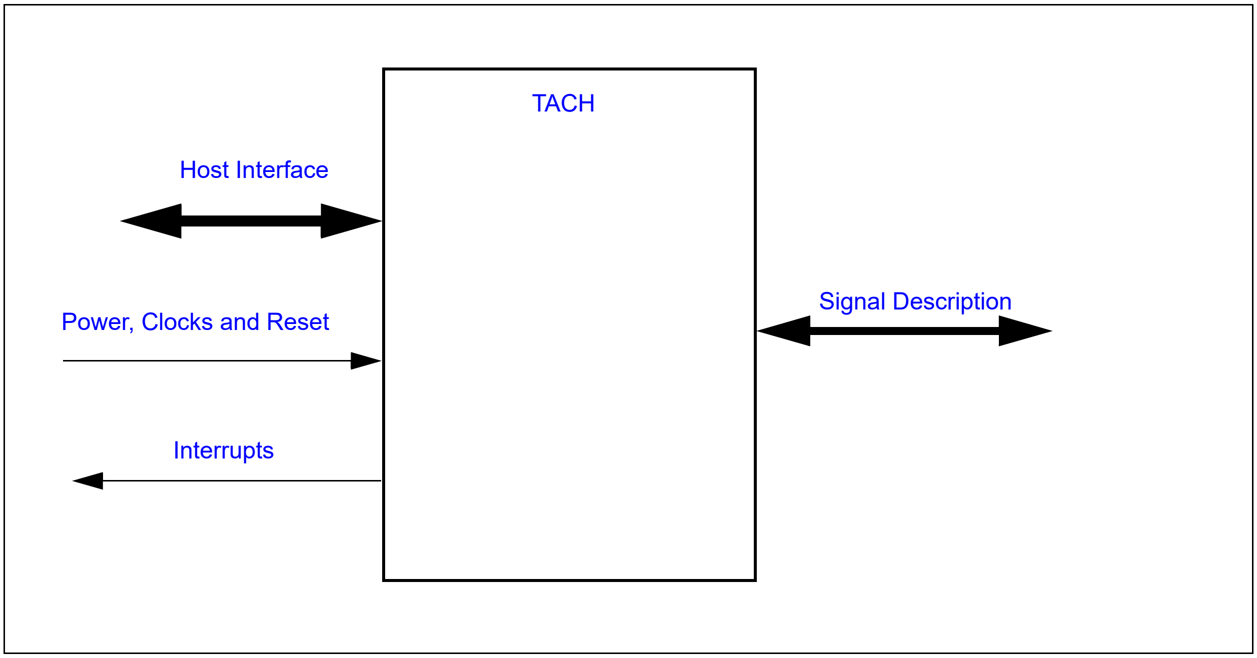
19.1 Introduction

This block monitors TACH output signals (or locked rotor signals) from various types of fans, and determines their speed.

19.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 19-1: I/O DIAGRAM OF BLOCK



19.3 Signal Description

TABLE 19-1: SIGNAL DESCRIPTION

Name	Direction	Description
TACH INPUT	Input	Tachometer signal from TACHx Pin.

19.4 Host Interface

The registers defined for the TACH are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

19.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

19.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

19.5.2 CLOCK INPUTS

Name	Description
100KHz	This is the clock input to the tachometer monitor logic. In Mode 1, the TACH is measured in the number of these clocks. This clock is derived from the main clock domain.

19.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

19.6 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 19-2: EC INTERRUPTS

Source	Description
TACH	This internal signal is generated from the OR'd result of the status events, as defined in the TACHx Status Register.

19.7 Low Power Modes

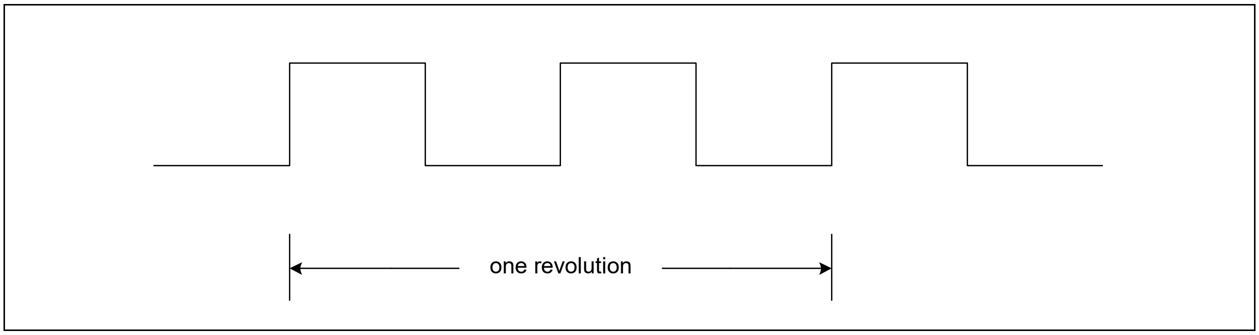
The TACH may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

19.8 Description

The TACH block monitors Tach output signals or locked rotor signals generated by various types of fans. These signals can be used to determine the speed of the attached fan. This block is designed to monitor fans at fan speeds from 100 RPMs to 30,000 RPMs.

Typically, these are DC brushless fans that generate (with each revolution) a 50% duty cycle, two-period square wave, as shown in Figure 19-2 below.

FIGURE 19-2: FAN GENERATED 50%DUTY CYCLE WAVEFORM



In typical systems, the fans are powered by the main power supply. Firmware may disable this block when it detects that the main power rail has been turned off by either clearing the <enable> TACH_ENABLE bit or putting the block to sleep via the supported Low Power Mode interface (see Low Power Modes).

19.8.1 MODES OF OPERATION

The Tachometer block supports two modes of operation. The mode of operation is selected via the TACH_READING_-MODE_SELECT bit.

19.8.1.1 Free Running Counter

In Mode 0, the Tachometer block uses the TACH input as the clock source for the internal TACH pulse counter (see TACHX_COUNTER). The counter is incremented when it detects a rising edge on the TACH input. In this mode, the firmware may periodically poll the TACHX_COUNTER field to determine the average speed over a period of time. The firmware must store the previous reading and the current reading to compute the number of pulses detected over a period of time. In this mode, the counter continuously increments until it reaches FFFFh. It then wraps back to 0000h and continues counting. The firmware must ensure that the sample rate is greater than the time it takes for the counter to wrap back to the starting point.

Note: Tach interrupts should be disabled in Mode 0.

19.8.1.2 Mode 1 -- Number of Clock Pulses per Revolution

In Mode 1, the Tachometer block uses its 100KHz clock input to measure the programmable number of TACH pulses. In this mode, the internal TACH pulse counter (TACHX_COUNTER) returns the value in number of 100KHz pulses per programmed number of TACH_EDGES. For fans that generate two square waves per revolution, these bits should be configured to five edges.

When the number of edges is detected, the counter is latched and the COUNT_READY_STATUS bit is asserted. If the COUNT_READY_INT_EN bit is set a TACH interrupt event will be generated.

19.8.2 OUT-OF-LIMIT EVENTS

The TACH Block has a pair of limit registers that may be configured to generate an event if the Tach indicates that the fan is operating too slow or too fast. If the <TACH reading> exceeds one of the programmed limits, the TACHx High Limit Register and the TACHx Low Limit Register, the bit TACH_OUT_OF_LIMIT_STATUS will be set. If the TACH_OUT_OF_LIMIT_STATUS bit is set, the Tachometer block will generate an interrupt event.

19.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the TACH Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 19-3: REGISTER SUMMARY

Offset	Register Name
00h	TACHx Control Register
04h	TACHx Status Register
08h	TACHx High Limit Register
0Ch	TACHx Low Limit Register

19.9.1 TACHX CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TACHX_COUNTER This 16-bit field contains the latched value of the internal Tach pulse counter, which may be configured by the Tach Reading Mode Select field to operate as a free-running counter or to be gated by the Tach input signal. If the counter is free-running (Mode 0), the internal Tach counter increments (if enabled) on transitions of the raw Tach input signal and is latched into this field every time it is incremented. The act of reading this field will not reset the counter, which rolls over to 0000h after FFFFh. The firmware will compute the delta between the current count reading and the previous count reading, to determine the number of pulses detected over a programmed period. If the counter is gated by the Tach input and clocked by 100KHz (Mode 1), the internal counter will be latched into the reading register when the programmed number of edges is detected or when the counter reaches FFFFh. The internal counter is reset to zero after it is copied into this register. **Note:** In Mode 1, a counter value of FFFFh means that the Tach did not detect the programmed number of edges in 655ms. A stuck fan can be detected by setting the TACHx High Limit Register to a number less than FFFFh. If the internal counter then reaches FFFFh, the reading register will be set to FFFFh and an out-of-limit interrupt can be sent to the EC.	R	00h	RESET_ SYS
15	TACH_INPUT_INT_EN 1=Enable Tach Input toggle interrupt from Tach block 0=Disable Tach Input toggle interrupt from Tach block	R/W	0b	RESET_ SYS
14	COUNT_READY_INT_EN 1=Enable Count Ready interrupt from Tach block 0=Disable Count Ready interrupt from Tach block	R/W	0b	RESET_ SYS
13	Reserved	RES	-	-
12:11	TACH_EDGES A Tach signal is a square wave with a 50% duty cycle. Typically, two Tach periods represents one revolution of the fan. A Tach period consists of three Tach edges. This programmed value represents the number of Tach edges that will be used to determine the interval for which the number of 100KHz pulses will be counted 11b=9 Tach edges (4 Tach periods) 10b=5 Tach edges (2 Tach periods) 01b=3 Tach edges (1 Tach period) 00b=2 Tach edges (1/2 Tach period)	R/W	00b	RESET_ SYS

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TACH_READING_MODE_SELECT 1=Counter is incremented on the rising edge of the 100KHz input. The counter is latched into the TACHX_COUNTER field and reset when the programmed number of edges is detected. 0=Counter is incremented when Tach Input transitions from low-to-high state (default)	R/W	0b	RESET_SYS
9	Reserved	RES	-	-
8	FILTER_ENABLE This filter is used to remove high frequency glitches from Tach Input. When this filter is enabled, Tach input pulses less than two 100KHz periods wide get filtered. 1=Filter enabled 0=Filter disabled (default) It is recommended that the Tach input filter always be enabled.	R/W	0b	RESET_SYS
7:2	Reserved	RES	-	-
1	TACH_ENABLE This bit gates the clocks into the block. When clocks are gated, the TACHx pin is tristated. When re-enabled, the internal counters will continue from the last known state and stale status events may still be pending. Firmware should discard any status or reading values until the reading value has been updated at least one time after the enable bit is set. 1=TACH Monitoring enabled, clocks enabled. 0=TACH Idle, clocks gated	R/W	0b	RESET_SYS
0	TACH_OUT_OF_LIMIT_ENABLE This bit is used to enable the TACH_OUT_OF_LIMIT_STATUS bit in the TACHx Status Register to generate an interrupt event. 1=Enable interrupt output from Tach block 0=Disable interrupt output from Tach block (default)	R/W	0b	RESET_SYS

19.9.2 TACHX STATUS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	COUNT_READY_STATUS This status bit is asserted when the Tach input changes state and when the counter value is latched. This bit remains cleared to '0' when the TACH_READING_MODE_SELECT bit in the TACHx Control Register is '0'. When the TACH_READING_MODE_SELECT bit in the TACHx Control Register is set to '1', this bit is set to ‘1’ when the counter value is latched by the hardware. It is cleared when written with a ‘1’. If COUNT_READY_INT_EN in the TACHx Control Register is set to 1, this status bit will assert the Tach Interrupt signal. 1=Reading ready 0=Reading not ready	R/WC	0b	RESET_SYS
2	TOGGLE_STATUS This bit is set when Tach Input changes state. It is cleared when written with a ‘1b’. If TACH_INPUT_INT_EN in the TACHx Control Register is set to ‘1b’, this status bit will assert the Tach Interrupt signal. 1=Tach Input changed state (this bit is set on a low-to-high or high-to-low transition) 0=Tach stable	R/WC	0b	RESET_SYS
1	TACH_PIN_STATUS This bit reflects the state of Tach Input. This bit is a read only bit that may be polled by the embedded controller. 1=Tach Input is high 0=Tach Input is low	R	0b	RESET_SYS
0	TACH_OUT_OF_LIMIT_STATUS This bit is set when the Tach Count value is greater than the high limit or less than the low limit. It is cleared when written with a ‘1b’. To disable this status event set the limits to their extreme values. If TACH_OUT_OF_LIMIT_ENABLE in the TACHx Control Register is set to 1’, this status bit will assert the Tach Interrupt signal. 1=Tach is outside of limits 0=Tach is within limits	R/WC	0b	RESET_SYS

Note:

- Some fans offer a Locked Rotor output pin that generates a level event if a locked rotor is detected. This bit may be used in combination with the Tach pin status bit to detect a locked rotor signal event from a fan.
- Tach Input may come up as active for Locked Rotor events. This would not cause an interrupt event because the pin would not toggle. Firmware must read the status events as part of the initialization process, if polling is not implemented.

19.9.3 TACHX HIGH LIMIT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	TACH_HIGH_LIMIT This value is compared with the value in the TACHX_COUNTER field. If the value in the counter is greater than the value programmed in this register, the TACH_OUT_OF_LIMIT_STATUS bit will be set. The TACH_OUT_OF_LIMIT_STATUS status event may be enabled to generate an interrupt to the embedded controller via the TACH_OUT_OF_LIMIT_ENABLE bit in the TACHx Control Register.	R/W	FFFFh	RESET_SYS

19.9.4 TACHX LOW LIMIT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	TACHX_LOW_LIMIT This value is compared with the value in the TACHX_COUNTER field of the TACHx Control Register. If the value in the counter is less than the value programmed in this register, the TACH_OUT_OF_LIMIT_STATUS bit will be set. The TACH_OUT_OF_LIMIT_STATUS status event may be enabled to generate an interrupt to the embedded controller via the TACH_OUT_OF_LIMIT_ENABLE bit in the TACHx Control Register To disable the TACH_OUT_OF_LIMIT_STATUS low event, program 0000h into this register.	R/W	0000h	RESET_SYS

20.0 PWM

20.1 Introduction

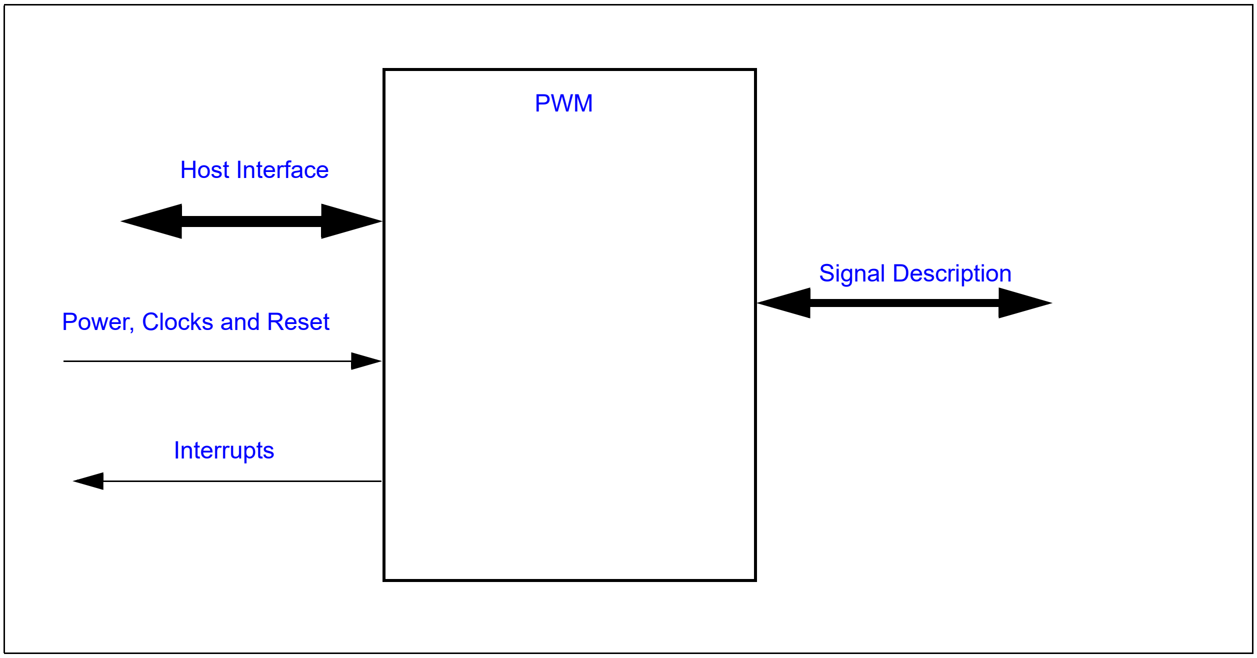
This block generates a PWM output that can be used to control 4-wire fans, blinking LEDs, and other similar devices. Each PWM can generate an arbitrary duty cycle output at frequencies from less than 0.1 Hz to 24 MHz.

The PWMx Counter ON Time registers and PWMx Counter OFF Time registers determine the operation of the PWM_OUTPUT signals. See Section 20.9.1, "PWMx Counter ON Time Register" and Section 20.9.2, "PWMx Counter OFF Time Register" for a description of the PWM_OUTPUT signals.

20.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 20-1: I/O DIAGRAM OF BLOCK



20.3 Signal Description

TABLE 20-1: SIGNAL DESCRIPTION

Name	Direction	Description
PWMx	OUTPUT	Pulse Width Modulated signal to PWMx pin.

20.4 Host Interface

The registers defined for the PWM Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

20.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

20.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

20.5.2 CLOCK INPUTS

Name	Description
48MHz	Clock input for generating high PWM frequencies, such as 15 kHz to 30 kHz.
100KHz	This is the clock input for generating low PWM frequencies, such as 10 Hz to 100 Hz.

20.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

20.6 Interrupts

The PWM block does not generate any interrupt events.

20.7 Low Power Modes

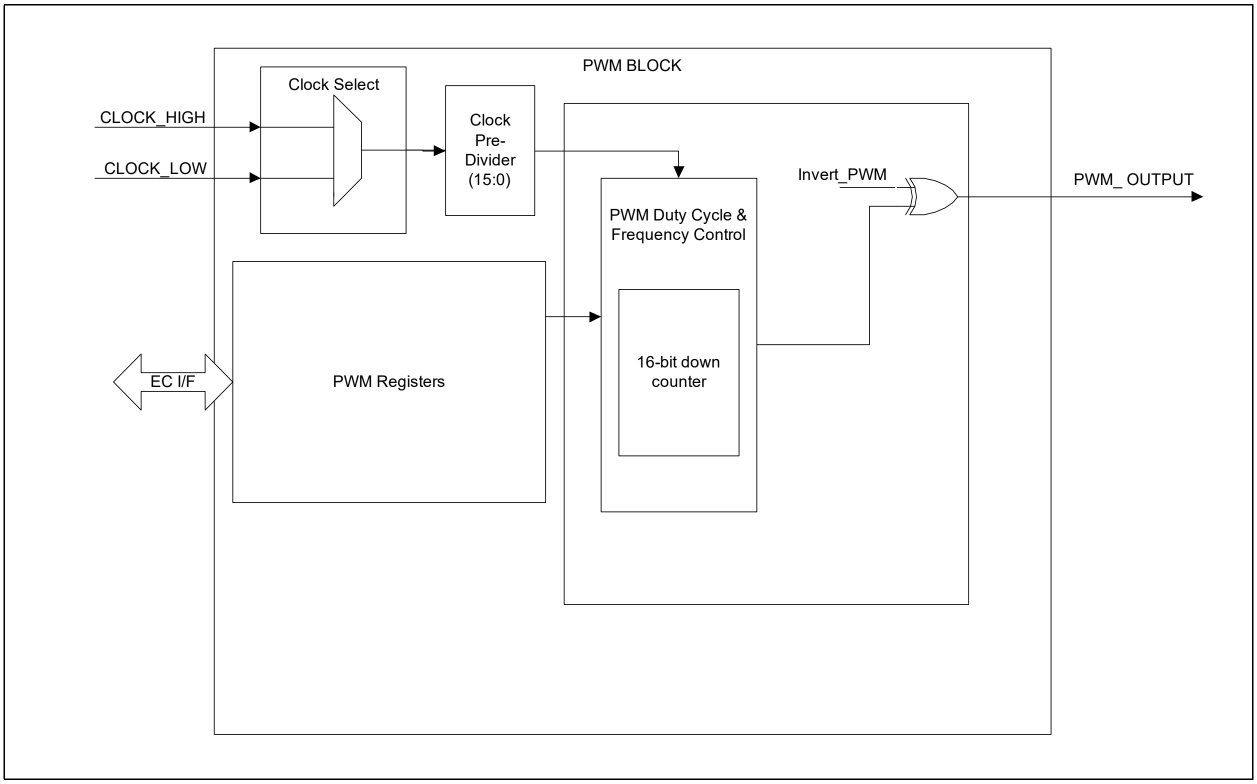
The PWM may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry. When the PWM is in the sleep state, the internal counters reset to 0 and the internal state of the PWM and the PWM_OUTPUT signal set to the OFF state.

20.8 Description

The PWM_OUTPUT signal is used to generate a duty cycle of specified frequency. This block can be programmed so that the PWM signal toggles the PWM_OUTPUT, holds it high, or holds it low. When the PWM is configured to toggle, the PWM_OUTPUT alternates from high to low at the rate specified in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register.

The following diagram illustrates how the clock inputs and registers are routed to the PWM Duty Cycle & Frequency Control logic to generate the PWM output.

FIGURE 20-2: BLOCK DIAGRAM OF PWM CONTROLLER



Note: In Figure 20-2, the 48MHz clock is represented as CLOCK_HIGH and the 100KHz clock is represented as CLOCK_LOW.

The PWM clock source to the PWM Down Counter, used to generate a duty cycle and frequency on the PWM, is determined through the Clock select[1] and Clock Pre-Divider[6:3] bits in the PWMx Configuration Register register.

The PWMx Counter ON/OFF Time registers determine both the frequency and duty cycle of the signal generated on PWM_OUTPUT as described below.

The PWM frequency is determined by the selected clock source and the total on and off time programmed in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers. The frequency is the time it takes (at that clock rate) to count down to 0 from the total on and off time.

The PWM duty cycle is determined by the relative values programmed in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers.

The PWM Frequency Equation and PWM Duty Cycle Equation are shown below.

EQUATION 20-1: PWM FREQUENCY EQUATION

$$\text{PWM Frequency} = \frac{1}{(PreDivisor+1)} \times \frac{(ClockSourceFrequency)}{((PWMCounterOnTime+1)+(PWMCounterOffTime+1))}$$

In this equation, the ClockSourceFrequency variable is the frequency of the clock source selected by the Clock Select bit in the PWMx Configuration Register, and PreDivisor is a field in the PWMx Configuration Register. The PWMCounterOnTime, PWMCounterOffTime are registers that are defined in Section 20.9, "EC Registers".

EQUATION 20-2: PWM DUTY CYCLE EQUATION

$$\text{PWM Duty Cycle} = \frac{(PWMCounterOnTime + 1)}{((PWMCounterOnTime + 1) + (PWMCounterOffTime + 1))}$$

The PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers should be accessed as 16-bit values.

20.8.1 PWM REGISTER UPDATES

The PWMx Counter ON Time Register and PWMx Counter OFF Time Register may be updated at any time. Values written into the two registers are kept in holding registers. The holding registers are transferred into the two user-visible registers when all four bytes have been written with new values and the internal counter completes the OFF time count. If the PWM is in the Full On state then the two user-visible registers are updated from the holding registers as soon as all four bytes have been written. Once the two registers have been updated the holding registers are marked empty. and all four bytes must again be written before the holding registers will be reloaded into the On Time Register and the Off Time Register. Reads of both registers return the current contents of the registers that are used to load the counter and not the holding registers.

20.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the PWM Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 20-2: REGISTER SUMMARY

Offset	Register Name
00h	PWMx Counter ON Time Register
04h	PWMx Counter OFF Time Register
08h	PWMx Configuration Register

20.9.1 PWMX COUNTER ON TIME REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	PWMX_COUNTER_ON_TIME This field determine both the frequency and duty cycle of the PWM signal. Setting this field to a value of *n* will cause the On time of the PWM to be *n*+1 cycles of the PWM Clock Source. When this field is set to zero and the PWMX_COUNTER_OFF_-TIME is not set to zero, the PWM_OUTPUT is held low (Full Off).	R/W	0000h	RESET_SYS

20.9.2 PWMX COUNTER OFF TIME REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	PWMX_COUNTER_OFF_TIME This field determine both the frequency and duty cycle of the PWM signal. Setting this field to a value of *n* will cause the Off time of the PWM to be *n*+1 cycles of the PWM Clock Source. When this field is set to zero, the PWM_OUTPUT is held high (Full On).	R/W	FFFFh	RESET_SYS

20.9.3 PWMX CONFIGURATION REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6:3	CLOCK_PRE_DIVIDER The Clock source for the 16-bit down counter (see PWMx Counter ON Time Register and PWMx Counter OFF Time Register) is determined by bit D1 of this register. The Clock source is then divided by the value of Pre-Divider+1 and the resulting signal determines the rate at which the down counter will be decremented. For example, a Pre-Divider value of 1 divides the input clock by 2 and a value of 2 divides the input clock by 3. A Pre-Divider of 0 will disable the Pre-Divider option.	R/W	0000b	RESET_SYS
2	INVERT 1=PWM_OUTPUT ON State is active low 0=PWM_OUTPUT ON State is active high	R/W	0b	RESET_SYS
1	CLOCK_SELECT This bit determines the clock source used by the PWM duty cycle and frequency control logic. 1=CLOCK_LOW 0=CLOCK_HIGH	R/W	0b	RESET_SYS
0	PWM_ENABLE When the PWM_ENABLE is set to 0 the internal counters are reset and the internal state machine is set to the OFF state. In addition, the PWM_OUTPUT signal is set to the inactive state as determined by the Invert bit. The PWMx Counter ON Time Register and PWMx Counter OFF Time Register are not affected by the PWM_ENABLE bit and may be read and written while the PWM enable bit is 0. 1=Enabled (default) 0=Disabled (gates clocks to save power)	R/W	0b	RESET_SYS

21.0 ANALOG TO DIGITAL CONVERTER

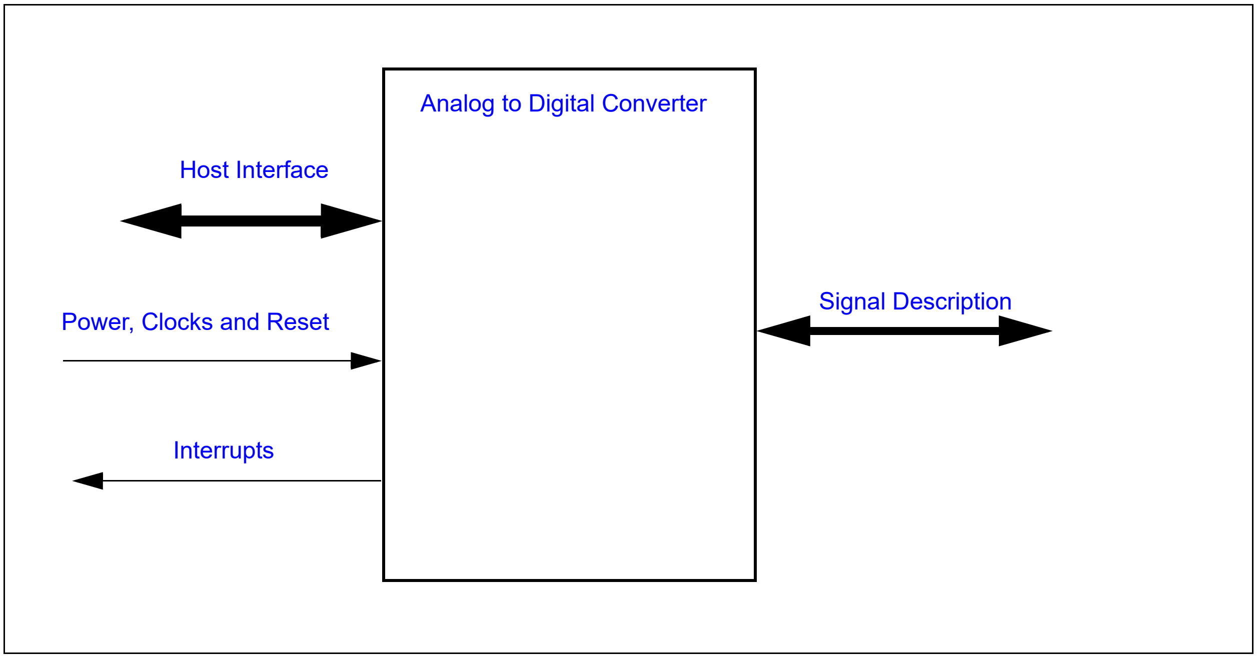
21.1 Introduction

This block is designed to convert external analog voltage readings into digital values. It consists of a single successive-approximation Analog-Digital Converter that can be shared among multiple inputs.

21.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 21-1: I/O DIAGRAM OF BLOCK



21.3 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 21-1: SIGNAL DESCRIPTION

Name	Direction	Description
ADC [7:0]	Input	ADC Analog Voltage Input from pins. **Note:** The ADC IP supports up to 12 channels. The number of channels implemented is package dependent. Refer to the Pin Chapter for the number of channels implemented in a package.
VREF_ADC	Input	ADC Reference Voltage input. ADC Reference Voltage. This pin must either be connected to a very accurate 3.3V reference or connected to the same VTR_ANALOG power supply that is powering the ADC logic.
VREF2_ADC	Input	ADC Reference Voltage input. ADC Reference Voltage can have 2 sources. 1. Internal Reference voltage sourced internal to the chip. This voltage will also be available on a GPIO pin for Thermistor reference voltage 2. External Reference voltage fed through GPIO pin

21.4 Host Interface

The registers defined for the ADC are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

21.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

21.5.1 POWER DOMAINS

TABLE 21-2: POWER SOURCES

Name	Description
VTR_CORE	This power well supplies power for the registers tn this block.
VTR_ANALOG	This power well supplies power for the analog circuitry in this block.

21.5.2 CLOCK INPUTS

TABLE 21-3: CLOCK INPUTS

Name	Description
48MHz	This clock signal is the master clock input to the ADC. This clock is internally divided to generate the ADC sampling clock. At 24MHz, the ADC does one channel conversion in 499.6nS for 12 bit resolution.

21.5.3 RESETS

TABLE 21-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.
SOFT_RESET	This is the Soft reset to the block and resets the Hardware in this block and does not affect the registers.

21.6 Interrupts

TABLE 21-5: EC INTERRUPTS

Source	Description
ADC_Single_Int	Interrupt signal from ADC controller to EC for Single-Sample ADC conversion.
ADC_Repeat_Int	Interrupt signal from ADC controller to EC for Repeated ADC conversion.

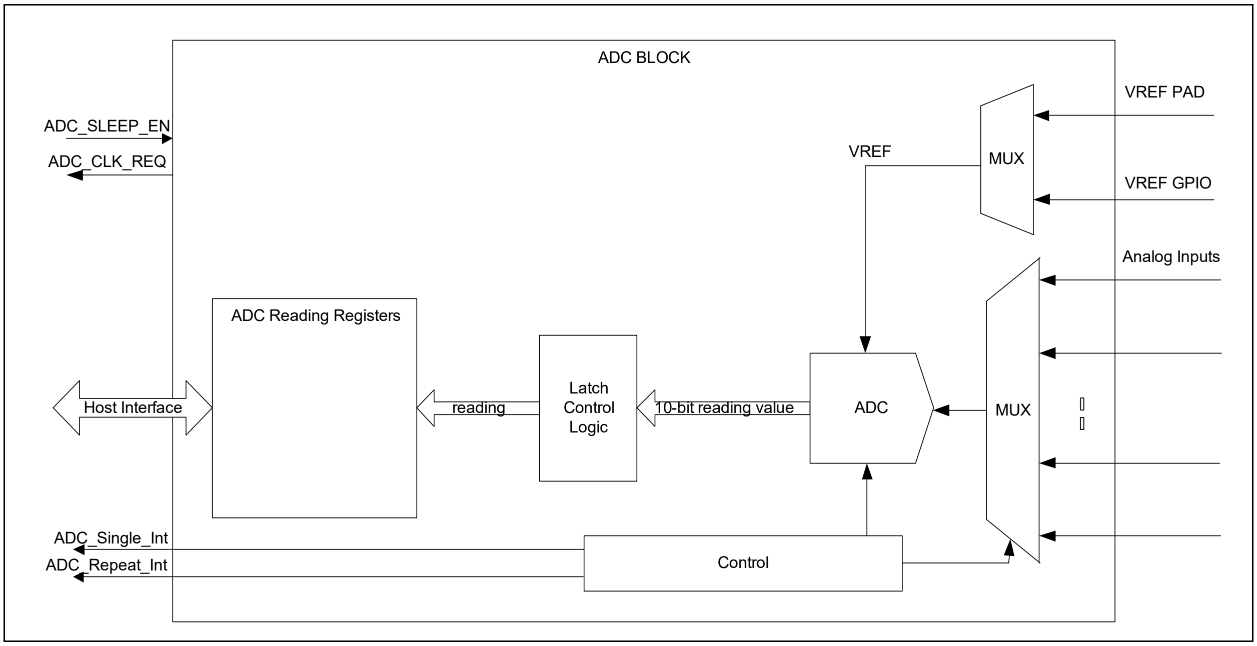
21.7 Low Power Modes

The ADC may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry.

The ADC is designed to conserve power when it is either sleeping or disabled. It is disabled via the ACTIVATE Bit and sleeps when the ADC_SLEEP_EN signal is asserted. The sleeping state only controls clocking in the ADC and does not power down the analog circuitry. For lowest power consumption, the ADC ACTIVATE bit must be set to ‘0.’

21.8 Description

FIGURE 21-2: ADC BLOCK DIAGRAM



The CEC1712 features a twelve channel successive approximation Analog to Digital Converter. The ADC architecture features excellent linearity and converts analog signals to 12 bit words. Conversion takes 499.6 nanoseconds per 12-bit word. The twelve channels are implemented with a single high speed ADC fed by a twelve input analog multiplexer. The multiplexer cycles through the twelve voltage channels, starting with the lowest-numbered channel and proceeding to the highest-number channel, selecting only those channels that are programmed to be active.

The input range on the voltage channels spans from 0V to the voltage reference. With an voltage reference of 3.3V, this provides resolutions of 3.2mV. The range can easily be extended with the aid of resistor dividers. The accuracy of any voltage reading depends on the accuracy and stability of the voltage reference input.

Note: The ADC pins are 3.3V tolerant.

Note: Transitions on ADC GPIOs are not permitted when Analog to Digital Converter readings are being taken.

Note: If GPIO and VREF2_ADC pins are shared and used as a GPIO noise can be injected into the ADC. Hence care should be taken in system design to make sure GPIOs doesn't switch when ADC is active.

The ADC conversion cycle starts either when the START_SINGLE bit in the ADC to set to 1 or when the ADC Repeat Timer counts down to 0. When the START_SINGLE is set to 1 the conversion cycle converts channels enabled by configuration bits in the ADC Single Register. When the Repeat Timer counts down to 0 the conversion cycle converts channels enabled by configuration bits in the ADC Repeat Register. When both the START_SINGLE bit and the Repeat Timer request conversions the START_SINGLE conversion is completed first.

Conversions always start with the lowest-numbered enabled channel and proceed to the highest-numbered enabled channel.

If software repeatedly sets Start_Single to 1 at a rate faster than the Repeat Timer count down interval, the conversion cycle defined by the ADC Repeat Register will not be executed.

21.8.1 REPEAT MODE

- Repeat Mode will start a conversion cycle of all ADC channels enabled by bits RPT_EN in the ADC Repeat Register. The conversion cycle will begin after a delay determined by WARM_UP_DELAY in SAR ADC Control Register and START_DELAY in the ADC Delay Register.Every channel that is enabled will be is converted in 500nS for 12 bit mode and 416.6nS for 10bit mode, for 24MHz internal reference clock. The conversion time formula is **Resolution * Sampling clock time period.**
- After all channels enabled by RPT_EN are complete, REPEAT_DONE_STATUS will be set to 1. The firmware will have to clear the REPEAT_DONE_STATUS bit.
- As long as START_REPEAT is 1 the ADC will repeatedly begin conversion cycles with a period defined by REPEAT_DELAY.
- If the delay period expires and a conversion cycle is already in progress because START_SINGLE was written with a 1, the cycle in progress will complete, followed immediately by a conversion cycle using RPT_EN to control the channel conversions.

21.8.2 SINGLE MODE

- The Single Mode conversion cycle will begin after WARM_UP_DELAY time. After all channels enabled by SINGLE_EN are complete, SINGLE_DONE_STATUS will be set to 1. The firmware will have to clear the SINGLE_DONE_STATUS bit.
- If START_SINGLE is written with a 1, while a conversion cycle is in progress because START_REPEAT is set, the current repeat conversion cycle will complete, followed immediately by a conversion cycle using SINGLE_EN to control the channel conversions.

21.8.3 APPLICATION NOTES

Please refer to white paper on “Accurate Temperature measurement using Thermistor” for details on how to use ADC for better than 1 degree C temperature measurement accuracy. Refer to FIGURE 21-3: ADC Reference Voltage Connection on page 225 for details of ADC reference voltage usage.

Note 1: ADC inputs require at least a 0.1 uF capacitor to filter glitches.

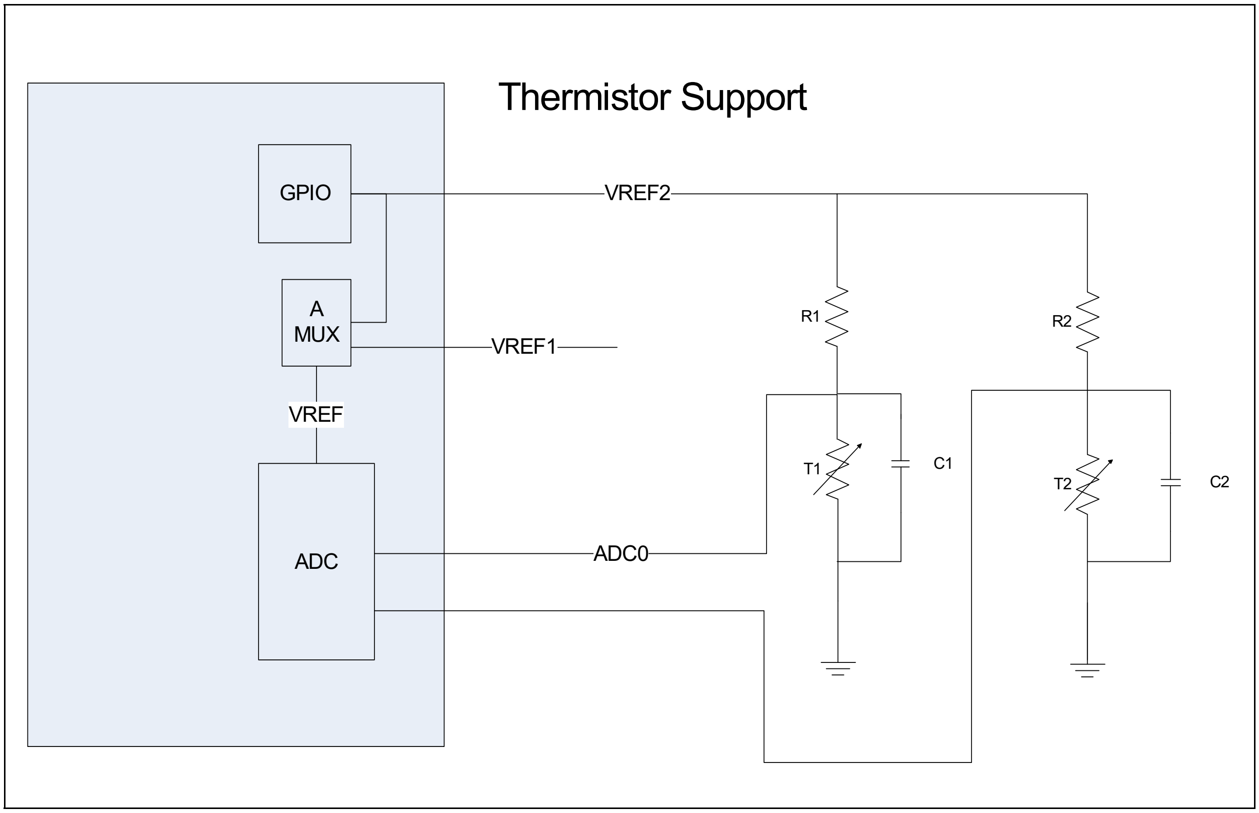
2: Use 1% Tolerance resistors with the ADC inputs.

3: It is recommended to use ADC sampling clock of 24MHz.

4: ADC sampling clock should not be configured to less than 3MHz.

5: Repeat_delay is dependent on the input impedance and sampling rate and will have to be tuned accordingly.

FIGURE 21-3: ADC REFERENCE VOLTAGE CONNECTION



21.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Analog to Digital Converter Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 21-6: REGISTER SUMMARY

Offset	Register Name
00h	ADC Control Register
04h	ADC Delay Register
08h	ADC Status Register
0Ch	ADC Single Register
10h	ADC Repeat Register
14h	ADC Channel Reading Registers 0
18h	ADC Channel Reading Registers 1
1Ch	ADC Channel Reading Registers 2
20h	ADC Channel Reading Registers 3
24h	ADC Channel Reading Registers 4
28h	ADC Channel Reading Registers 5
2Ch	ADC Channel Reading Registers 6
30h	ADC Channel Reading Registers 7
7Ch	ADC Configuration Register

TABLE 21-6: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
80h	It is recommended to operate at ADC sampling clock = 24MHz; Do not set the ADC sampling clock to be less than 3MHz.
84h	VREF Control Register
88h	SAR ADC Control Register

21.9.1 ADC CONTROL REGISTER

The ADC Control Register is used to control the behavior of the Analog to Digital Converter.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	SINGLE_DONE_STATUS This bit is cleared when it is written with a 1. Writing a 0 to this bit has no effect. This bit can be used to generate an EC interrupt. 1= ADC single-sample conversion is completed. This bit is set to 1 when conversion completes for all enabled channels in the single conversion cycle 0= ADC single-sample conversion is not complete. This bit is cleared whenever the software writes a 1b to this bit. **Note:** Only firmware is able to clear SINGLE_DONE_STATUS and REPEAT_DONE_STATUS status bits by writing a 1 to these bits, even when multiple repeat_done or single_-done events occurs before firmware services the interrupt. **Note:** This bit is not self clearing bit.	R/WC	0h	RESET_SYS
6	REPEAT_DONE_STATUS This bit is cleared when it is written with a 1. Writing a 0 to this bit has no effect. This bit can be used to generate an EC interrupt. 1= ADC repeat-sample conversion is completed. This bit is set to 1 when all enabled channels in a repeating conversion cycle complete 0= ADC repeat-sample conversion is not complete.This bit is cleared whenever the software writes to this bit to clear it. **Note:** Only firmware is able to clear SINGLE_DONE_STATUS and REPEAT_DONE_STATUS status bits by writing a 1 to these bits, even when multiple repeat_done or single_-done events occurs before firmware services the interrupt. **Note:** This bit is not self clearing bit.	R/WC	0h	RESET_SYS
5	Reserved	RES	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	SOFT_RESET 1=writing one causes a reset of the ADC block hardware (not the registers) 0=writing zero takes the ADC block out of reset	R/W	0h	RESET_SYS
3	POWER_SAVER_DIS 1=Power saving feature is disabled **Note:** 0=Power saving feature is enabled. The Analog to Digital Converter controller powers down the ADC between conversion sequences.	R/W	0h	RESET_SYS
2	START_REPEAT 1=The ADC Repeat Mode is enabled. This setting will start a conversion cycle of all ADC channels enabled by bits RPT_EN in the ADC Repeat Register. 0=The ADC Repeat Mode is disabled. Note: This setting will not terminate any conversion cycle in process, but will clear the Repeat Timer and inhibit any further periodic conversions.	R/W	0h	RESET_SYS
1	START_SINGLE 1=The ADC Single Mode is enabled. This setting starts a single conversion cycle of all ADC channels enabled by bits SINGLE_EN in the ADC Single Register. 0=The ADC Single Mode is disabled. This bit is self-clearing	R/W	0h	RESET_SYS
0	ACTIVATE 1=ADC block is enabled for operation. START_SINGLE or START_REPEAT can begin data conversions by the ADC. Note: A reset pulse is sent to the ADC core when this bit changes from 0 to 1. 0=The ADC is disabled and placed in its lowest power state. Note: Any conversion cycle in process will complete before the block is shut down, so that the reading registers will contain valid data but no new conversion cycles will begin.	R/W	0h	RESET_SYS

21.9.2 ADC DELAY REGISTER

The ADC Delay register determines the delay from setting START_REPEAT in the ADC Control Register and the start of a conversion cycle. This register also controls the interval between conversion cycles in repeat mode.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	REPEAT_DELAY This field determines the interval between conversion cycles when START_REPEAT is 1. The delay is in units of 40µs. A value of 0 means no delay between conversion cycles, and a value of 0xFFFF means a delay of 2.6 seconds. This field has no effect when START_SINGLE is written with a 1. **Note 1:** The REPEAT_DELAY is the delay before the start of each successive repeat cycle (not the first cycle. START_DELAY will be used for the first cycle) when the ADC is in low power state and the only after this delay the enable to the actual ADC block is asserted. The delay is also relative to the source being monitored. for eg. for a 10K impedance the recommended REPEAT_DELAY is 0x10 **2:** If the ADC sampling clock is 24MHz, the default REPEAT_DELAY of 0 will work, for lower sampling rate the value will change	R/W	0000h	RESET_SYS
15:0	START_DELAY This field determines the starting delay before a conversion cycle is begun when START_REPEAT is written with a 1. The delay is in units of 40µs. A value of 0 means no delay before the start of a conversion cycle, and a value of 0xFFFF means a delay of 2.6 seconds. This field has no effect when START_SINGLE is written with a 1. **Note:** The START_DELAY is the delay before the start of new repeat cycle when the ADC is disabled and only after this delay the enable to the actual ADC core is asserted.	R/W	0000h	RESET_SYS

21.9.3 ADC STATUS REGISTER

The ADC Status Register indicates whether the ADC has completed a conversion cycle.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	ADC_CH_STATUS All bits are cleared by being written with a ‘1’. 1=conversion of the corresponding ADC channel is complete 0=conversion of the corresponding ADC channel is not complete For enabled single cycles, the SINGLE_DONE_STATUS bit in the ADC Control Register is also set after all enabled channel conversion are done; for enabled repeat cycles, the REPEAT_DONE_STATUS in the ADC Control Register is also set after all enabled channel conversion are done.	R/WC	00h	RESET_SYS

21.9.4 ADC SINGLE REGISTER

The ADC Single Register is used to control which ADC channel is captured during a Single-Sample conversion cycle initiated by the START_SINGLE bit in the ADC Control Register.

> **Note:** Do not change the bits in this register in the middle of a conversion cycle to insure proper operation.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	SINGLE_EN Each bit in this field enables the corresponding ADC channel when a single cycle of conversions is started when the START_SINGLE bit in the ADC Control Register is written with a 1. 1=single cycle conversions for this channel are enabled 0=single cycle conversions for this channel are disabled	R/W	0h	RESET_ SYS

21.9.5 ADC REPEAT REGISTER

The ADC Repeat Register is used to control which ADC channels are captured during a repeat conversion cycle initiated by the START_REPEAT bit in the ADC Control Register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	RPT_EN Each bit in this field enables the corresponding ADC channel for each pass of the Repeated ADC Conversion that is controlled by bit START_REPEAT in the ADC Control Register. 1=repeat conversions for this channel are enabled 0=repeat conversions for this channel are disabled	R/W	00h	RESET_ SYS

21.9.6 ADC CHANNEL READING REGISTERS

All ADC channels return their results into a 32-bit reading register. In each case the low 10/12 bits of the reading register return the result of the Analog to Digital conversion and the upper 22/20 bits return 0. Table 21-6, "Register Summary" shows the addresses of all the reading registers.

> **Note:** The ADC Channel Reading Registers access require single 16, or 32 bit reads; i.e., two 8 bit reads will not provide data coherency.

21.9.7 ADC CONFIGURATION REGISTER

Offset	7Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TEST	R	-	-
15:8	ADC_CLK_HIGH_TIME High Time Count ADC Clock: Programmable from 1 to 255. 0 is not used. **Note:** The High Time Count must be programmed to be equal to the Low Time Count (must be programmed to 50% duty cycle).	R/W	01h	RESET_SYS
7:0	ADC_CLK_LOW_TIME Low Time Count ADC Clock: Programmable from 1 to 255. 0 is not used. The High Time Count must be programmed to be equal to the Low Time Count (must be programmed to 50% duty cycle).	R/W	01h	RESET_SYS

Note: It is recommended to operate at ADC sampling clock = 24MHz; Do not set the ADC sampling clock to be less than 3MHz.

21.9.8 VREF CHANNEL REGISTER

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:22	VREF Select for Channel 11 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
21:20	VREF Select for Channel 10 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
19:18	VREF Select for Channel 9 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
17:16	VREF Select for Channel 8 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:14	VREF Select for Channel 7 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
13:12	VREF Select for Channel 6 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
11:10	VREF Select for Channel 5 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
9:8	VREF Select for Channel 4 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
7:6	VREF Select for Channel 3 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
5:4	VREF Select for Channel 2 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
3:2	VREF Select for Channel 1 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
1:0	VREF Select for Channel 0 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS

21.9.9 VREF CONTROL REGISTER

Offset	84h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:30	VREF Select Status These bits show the VREF selected at this time of reading the register.	R	0h	RESET_SYS
29	VREF_PAD_CTL This is the VREF Pad Control 0 = Leave unused pad floating 1 = Drive unused pad low	R/W	0h	RESET_SYS
28:16	VREF Switch Delay This is the time delay required to switch VREF selects	R/W	0h	RESET_SYS
15:0	VREF Charge Delay This is the time delay required to charge the external VREF capacitor	R/W	0h	RESET_SYS

21.9.10 SAR ADC CONTROL REGISTER

Offset	88h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	Reserved	RES	-	-
16:7	WARM_UP_DELAY This is the warm up time delay required for ADC. The delay is in terms of number of ADC clock cycles.	R/W	202h	RESET_SYS
6-4	Reserved	RES	-	-
3	SHIFT_DATA Right justify ADC output data 0 = ADC_DOUT is not shifted and lower bits are 0 1 = ADC_DOUT is shifted right following resolution selection	R/W	0h	RESET_SYS
2:1	SEL_RES These bits define the SAR ADC resolution 00b = Reserved 01b = Reserved 10b = 10 bit resolution 11b = 12 bit resolution	R/W	3h	RESET_SYS
0	SELDIFF This bit define the single ended / differential mode of ADC operation 0 = ADC is enabled for single ended input operation 1 = ADC is enabled for differential mode input operation	R/W	0h	RESET_SYS

22.0 BLINKING/BREATHING LED

22.1 Introduction

LEDs are used in computer applications to communicate internal state information to a user through a minimal interface. Typical applications will cause an LED to blink at different rates to convey different state information. For example, an LED could be full on, full off, blinking at a rate of once a second, or blinking at a rate of once every four seconds, in order to communicate four different states.

As an alternative to blinking, an LED can “breathe”, that is, oscillate between a bright state and a dim state in a continuous, or apparently continuous manner. The rate of breathing, or the level of brightness at the extremes of the oscillation period, can be used to convey state information to the user that may be more informative, or at least more novel, than traditional blinking.

The blinking/breathing hardware is implemented using a PWM. The PWM can be driven either by the Main system clock or by a 32.768 KHz clock input. When driven by the Main system clock, the PWM can be used as a standard 8-bit PWM in order to control a fan. When used to drive blinking or breathing LEDs, the 32.768 KHz clock source is used.

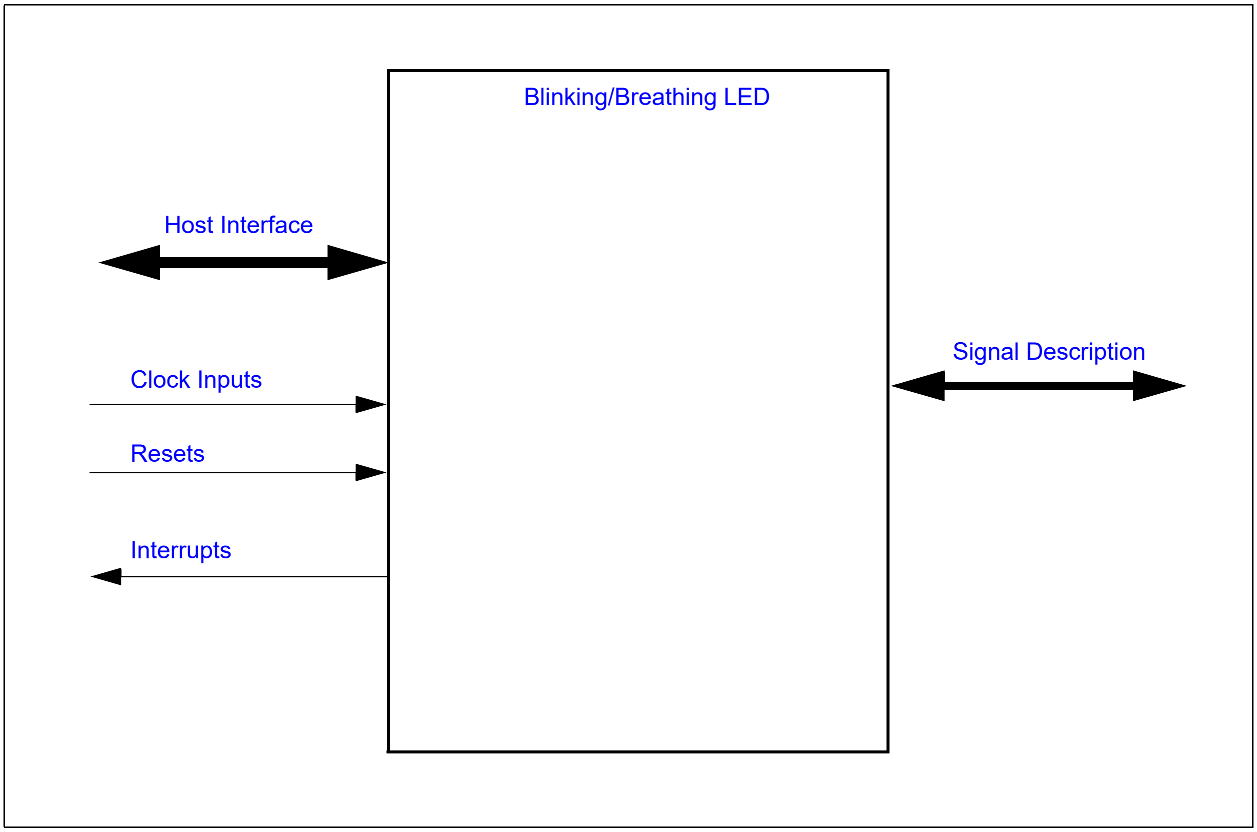
Features:

- Each PWM independently configurable
- Each PWM configurable for LED blinking and breathing output
- Highly configurable breathing rate from 60ms to 1min
- Non-linear brightness curves approximated with 8 piece wise-linear segments
- All LED PWMs can be synchronized
- Each PWM configurable for 8-bit PWM support
- Multiple clock rates
- Configurable Watchdog Timer

22.2 Interface

This block is designed to drive a pin on the pin interface and to be accessed internally via a registered host interface.

FIGURE 22-1: I/O DIAGRAM OF BLOCK



22.3 Signal Description

Name	Direction	Description
LEDx	Output	PWM LED Output[a] By default, the LEDx pin is configured to be active high: when the LED is configured to be fully on, the pin is driving high. When the LED is configured to be fully off, the pin is low. If firmware requires the Blinking/Breathing PWM to be active low, the Polarity bit in the GPIO Pin Control Register associated with the LED can be set to 1, which inverts the output polarity.

[a] Refer to the Table 1-1, "CEC1712 Feature list" table to know the number of LED pins available in the chip.

22.4 Host Interface

The blinking/breathing PWM block is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

22.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

22.5.1 POWER DOMAINS

Name	Description
VTR_CORE	Main power. The source of main power for the device is system dependent.

22.5.2 CLOCK INPUTS

Name	Description
32KHz	32.768 KHz clock
48MHz	Main system clock

22.5.3 RESETS

Name	Description
RESET_SYS	This reset signal resets all the logic and register in this block.
RESET	This reset signal, resets the PWM registers to their default values.

22.6 Interrupts

Each PWM can generate an interrupt. The interrupt is asserted for one Main system clock period whenever the PWM WDT times out. The PWM WDT is described in Section 22.8.3.1, "PWM WDT".

Source	Description
PWM_WDT	PWM watchdog time out

22.7 Low Power Mode

The Blinking/Breathing LED may be put into a low power mode by the chip-level power, clocks, and reset (PCR) circuitry. The low power mode is only applicable when the Blinking/Breathing PWM is operating in the General Purpose PWM mode. When the low speed clock mode is selected, the blinking/breathing function continues to operate, even when the 48MHz is stopped. Low power mode behavior is summarized in the following table:

TABLE 22-1: LOW POWER MODE BEHAVIOR

CLOCK_SOURCE	CONTROL	Mode	Low Power Mode	Description
X	'00'b	PWM 'OFF'	Yes	32.768 KHz clock is required.
X	'01'b	Breathing	Yes	
1	'10'b	General Purpose PWM	No	Main system clock is required, even when a sleep command to the block is asserted.
0	'10'b	Blinking	Yes	32.768 KHz clock is required.
X	'11'b	PWM 'ON'	Yes	

Note: In order for the CEC1712 to enter its.Heavy Sleep state, the SLEEP_ENABLE input for all Blinking/Breathing PWM instances must be asserted, even if the PWMs are configured to use the low speed clock.

22.8 Description

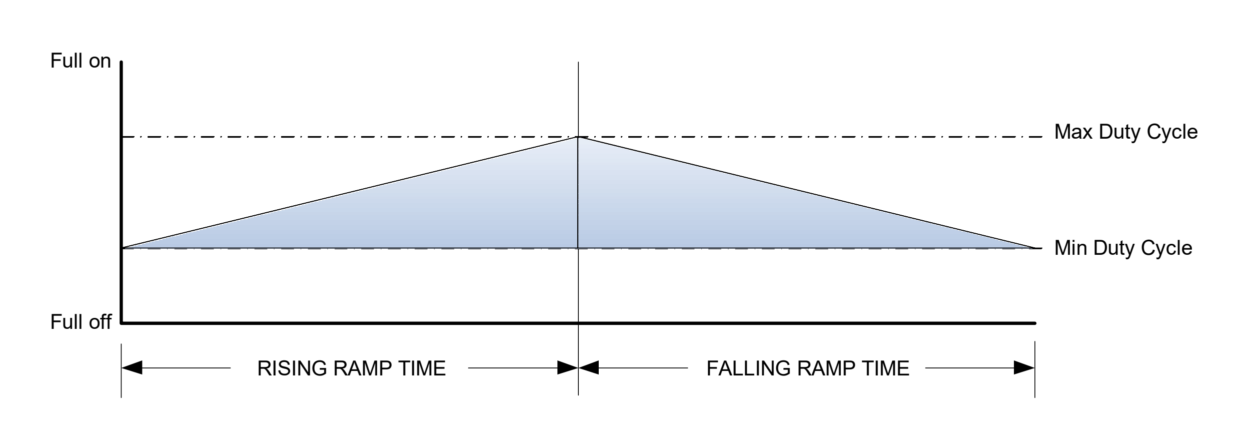
22.8.1 BREATHING

If an LED blinks rapidly enough, the eye will interpret the light as reduced brightness, rather than a blinking pattern. Therefore, if the blinking period is short enough, modifying the duty cycle will set the apparent brightness, rather than a blinking rate. At a blinking rate of 128Hz or greater, almost all people will perceive a continuous light source rather than an intermittent pattern.

Because making an LED appear to breathe is an aesthetic effect, the breathing mechanism must be adjustable or customers may find the breathing effect unattractive. There are several variables that can affect breathing appearance, as described below.

The following figure illustrates some of the variables in breathing:

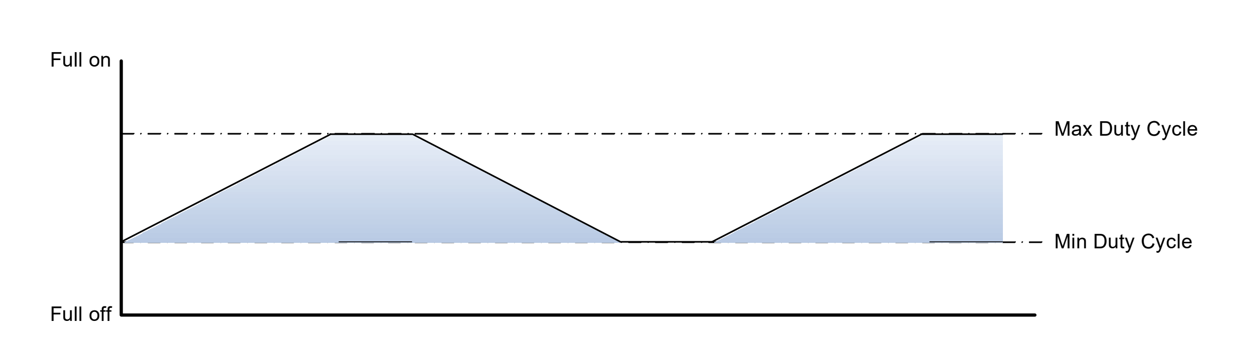
FIGURE 22-2: BREATHING LED EXAMPLE



The breathing range of and LED can range between full on and full off, or in a range that falls within the full-on/full-off range, as shown in this figure. The ramp time can be different in different applications. For example, if the ramp time was 1 second, the LED would appear to breathe quickly. A time of 2 seconds would make the LED appear to breathe more leisurely.

The breathing pattern can be clipped, as shown in the following figure, so that the breathing effect appears to pause at its maximum and minimum brightnesses:

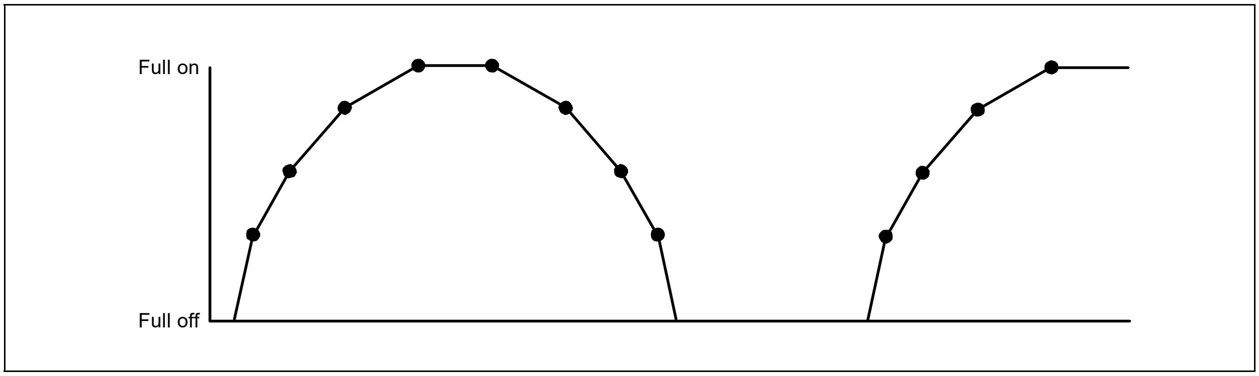
FIGURE 22-3: CLIPPING EXAMPLE



The clipping periods at the two extremes can be adjusted independently, so that for example an LED can appear to breathe (with a short delay at maximum brightness) followed by a longer “resting” period (with a long delay at minimum brightness).

The brightness can also be changed in a non-linear fashion, as shown in the following figure:

FIGURE 22-4: EXAMPLE OF A SEGMENTED CURVE



In this figure, the rise and fall curves are implemented in 4 linear segments and are the rise and fall periods are symmetric.

The breathing mode uses the 32.768 KHz clock for its time base.

22.8.2 BLINKING

When configured for blinking, a subset of the hardware used in breathing is used to implement the blinking function. The PWM (an 8-bit accumulator plus an 8-bit duty cycle register) drives the LED directly. The Duty Cycle register is programmed directly by the user, and not modified further. The PWM accumulator is configured as a simple 8-bit up counter. The counter uses the 32.768 KHz clock, and is pre-scaled by the Delay counter, to slow the PWM down from the 128Hz provided by directly running the PWM on the 32.768 KHz clock.

With the pre-scaler, the blink rate of the LED could be as fast as 128Hz (which, because it is blinking faster than the eye can distinguish, would appear as a continuous level) to 0.03125Hz (that is, with a period of 7.8ms to 32 seconds). Any duty cycle from 0% (0h) to 100% (FFh) can be configured, with an 8-bit precision. An LED with a duty cycle value of 0h will be fully off, while an LED with a duty cycle value of FFh will be fully on.

In Blinking mode the PWM counter is always in 8-bit mode.

Table 22-2, "LED Blink Configuration Examples" shows some example blinking configurations:

TABLE 22-2: LED BLINK CONFIGURATION EXAMPLES

Prescale	Duty Cycle	Blink Frequency	Blink
000h	00h	128Hz	full off
000h	FFh	128Hz	full on
001h	40h	64Hz	3.9ms on, 11.5ms off
003h	80h	32Hz	15.5ms on, 15.5ms off
07Fh	20h	1Hz	125ms on, 0.875s off
0BFh	16h	0.66Hz	125ms on, 1.375s off
0FFh	10h	0.5Hz	125ms on, 1.875s off
180h	0Bh	0.33Hz	129ms on, 2.875s off
1FFh	40h	0.25Hz	1s on, 3s off

The Blinking and General Purpose PWM modes share the hardware used in the breathing mode. The Prescale value is derived from the LD field of the LED_DELAY register and the Duty Cycle is derived from the MIN field of the LED_LIMITS register.

TABLE 22-3: BLINKING MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(32KHz frequency) /(PRESCALE + 1)/256
‘H’ Width	Seconds	(1/Frequency) x (DutyCycle/256)
‘L’ Width	Seconds	(1/Frequency) x ((1-DutyCycle)/256)

22.8.3 GENERAL PURPOSE PWM

When used in the Blinking configuration with the 48MHz, the LED module can be used as a general-purpose programmable Pulse-Width Modulator with an 8-bit programmable pulse width. It can be used for fan speed control, sound volume, etc. With the 48MHz source, the PWM frequency can be configured in the range shown in Table 22-4.

TABLE 22-4: PWM CONFIGURATION EXAMPLES

Prescale	PWM Frequency
000h	187.5 KHz
001h	94 KHz
003h	47 KHz
006h	26.8 KHz
00Bh	15.625 KHz
07Fh	1.46 KHz
1FFh	366 Hz
FFFh	46 Hz

TABLE 22-5: GENERAL PURPOSE PWM MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(48MHz frequency) / (PRESCALE + 1) / 256
‘H’ Width	Seconds	(1/Frequency) x (DutyCycle/256)
‘L’ Width	Seconds	(1/Frequency) x (256 - DutyCycle)

22.8.3.1 PWM WDT

When the PWM is configured as a general-purpose PWM (in the Blinking configuration with the Main system clock), the PWM includes a Watch Dog Timer (WDT). The WDT consists of an internal 8-bit counter and an 8-bit reload value (the field WDTLD in LED Configuration Register). The internal counter is loaded with the reset value of WDTLD (14h, or 4 seconds) on system RESET_SYS and loaded with the contents of WDTLD whenever either the LED Configuration Register register is written or the MIN byte in the LED Limits Register register is written (the MIN byte controls the duty cycle of the PWM).

Whenever the internal counter is non-zero, it is decremented by 1 for every tick of the 5 Hz clock. If the counter decrements from 1 to 0, a WDT Terminal Count causes an interrupt to be generated and reset sets the CONTROL bit in the LED Configuration Register to 3h, which forces the PWM to be full on. No other PWM registers or fields are affected.

If the 5 Hz clock halts, the watchdog timer stops decrementing but retains its value, provided the device continues to be powered. When the 5 Hz clock restarts, the watchdog counter will continue decrementing where it left off.

Setting the WDTLD bits to 0 disables the PWM WDT. Other sample values for WDTLD are:

01h = 200 ms

02h = 400 ms

03h = 600 ms

04h = 800 ms

...

14h = 4seconds

FFh = 51 seconds

22.9 Implementation

In addition to the registers described in Section 22.10, "EC Registers", the PWM is implemented using a number of components that are interconnected differently when configured for breathing operation and when configured for blinking/PWM operation.

22.9.1 BREATHING CONFIGURATION

The **PSIZE** parameter can configure the PWM to one of three modes: 8-bit, 7-bit and 6-bit. The **PERIOD CTR** counts ticks of its input clock. In 8-bit mode, it counts from 0 to 255 (that is, 256 steps), then repeats continuously. In this mode, a full cycle takes 7.8ms (128Hz). In 7-bit mode it counts from 0 to 127 (128 steps), and a full cycle takes 3.9ms (256Hz). In 6-bit mode it counts from 0 to 63 (64 steps) and a full cycle takes 1.95ms (512Hz).

The output of the LED circuit is asserted whenever the **PERIOD CTR** is less than the contents of the **DUTY CYCLE** register. The appearance of breathing is created by modifying the contents of the **DUTY CYCLE** register in a continuous manner. When the LED control is off the internal counters and registers are all reset to 0 (i.e. after a write setting the RESET bit in the LED Configuration Register Register.) Once enabled, the **DUTY CYCLE** register is increased by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. Once the duty cycle reaches its maximum value (determined by the field MAX), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the **DUTY CYCLE** register is decreased, again by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. When the duty cycle then falls at or below the minimum value (determined by the field MIN), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the cycle repeats, with the duty cycle oscillating between MIN and MAX.

The rising and falling ramp times as shown in Figure 22-2, "Breathing LED Example" can be either symmetric or asymmetric depending on the setting of the SYMMETRY bit in the LED Configuration Register Register. In Symmetric mode the rising and falling ramp rates have mirror symmetry; both rising and falling ramp rates use the same (all) 8 segments fields in each of the following registers (see Table 22-6): the LED Update Stepsize Register register and the LED Update Interval Register register. In Asymmetric mode the rising ramp rate uses 4 of the 8 segments fields and the falling ramp rate uses the remaining 4 of the 8 segments fields (see Table 22-6).

The parameters MIN, MAX, HD, LD and the 8 fields in LED_STEP and LED_INT determine the brightness range of the LED and the rate at which its brightness changes. See the descriptions of the fields in Section 22.10, "EC Registers", as well as the examples in Section 22.9.3, "Breathing Examples" for information on how to set these fields.

TABLE 22-6: SYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 22-3, "Clipping Example"	Duty Cycle	Segment Index	Symmetric Mode Register Fields Utilized	
X	000xxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
X	001xxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
X	010xxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
X	011xxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
X	100xxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
X	101xxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
X	110xxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
X	111xxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]

Note: In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]

TABLE 22-7: ASYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 22-3, "Clipping Example"	Duty Cycle	Segment Index	Asymmetric Mode Register Fields Utilized	
Rising	00xxxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
Rising	01xxxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
Rising	10xxxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
Rising	11xxxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
falling	00xxxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
falling	01xxxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
falling	10xxxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
falling	11xxxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]

Note: In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 22-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

22.9.2 BLINKING CONFIGURATION

The Delay counter and the PWM counter are the same as in the breathing configuration, except in this configuration they are connected differently. The Delay counter is clocked on either the 32.768 KHz clock or the Main system clock, rather than the output of the PWM. The PWM counter is clocked by the zero output of the Delay counter, which functions as a prescalar for the input clocks to the PWM. The Delay counter is reloaded from the LD field of the LED_DELAY register. When the LD field is 0 the input clock is passed directly to the PWM counter without prescaling. In Blinking/PWM mode the PWM counter is always 8-bit, and the PSIZE parameter has no effect.

The frequency of the PWM pulse waveform is determined by the formula:

$$f_{PWM} = \frac{f_{clock}}{(256 \times (LD + 1))}$$

where f_{PWM} is the frequency of the PWM, f_{clock} is the frequency of the input clock (32.768 KHz clock or Main system clock) and LD is the contents of the LD field.

Note: At a duty cycle value of 00h (in the MIN register), the LED output is fully off. At a duty cycle value of 255h, the LED output is fully on. Alternatively, In order to force the LED to be fully on, firmware can set the CONTROL field of the Configuration register to 3 (always on).

The other registers in the block do not affect the PWM or the LED output in Blinking/PWM mode.

22.9.3 BREATHING EXAMPLES

22.9.3.1 Linear LED brightness change

In this example, the brightness of the LED increases and diminishes in a linear fashion. The entire cycle takes 5 seconds. The rise time and fall time are 1.6 seconds, with a hold time at maximum brightness of 200ms and a hold time at minimum brightness of 1.6 seconds. The LED brightness varies between full off and full on. The PWM size is set to 8-bit, so the time unit for adjusting the PWM is approximately 8ms. The registers are configured as follows:

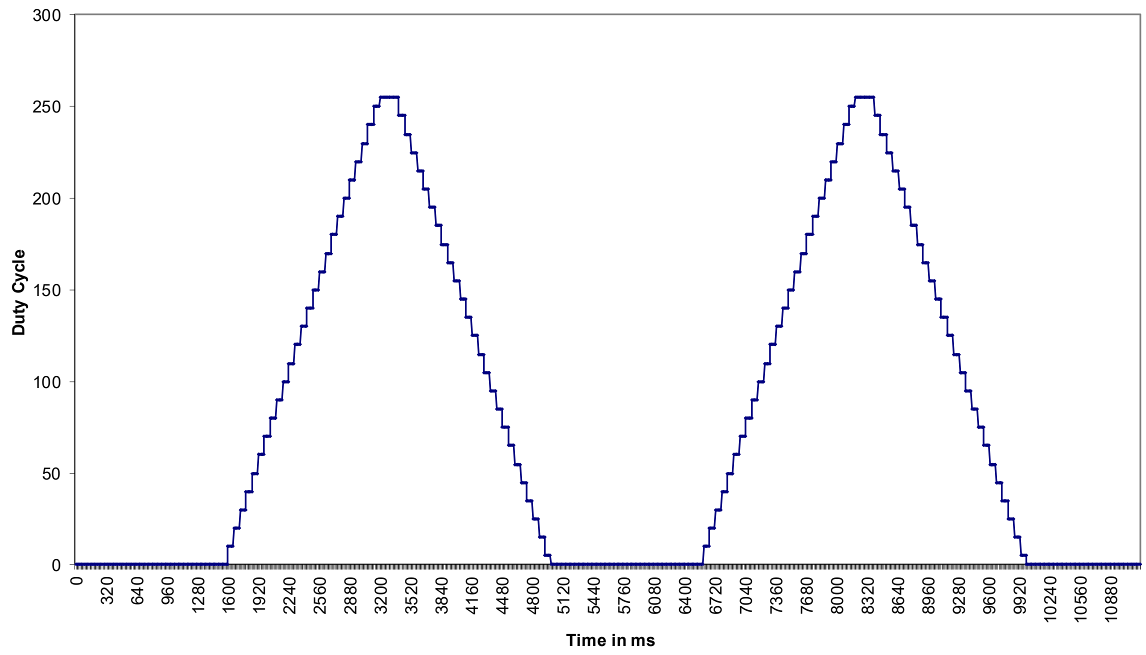
TABLE 22-8: LINEAR EXAMPLE CONFIGURATION

Field	Value
PSIZE	8-bit
MAX	255
MIN	0
HD	25 ticks (200ms)
LD	200 ticks (1.6s)

TABLE 22-8: LINEAR EXAMPLE CONFIGURATION (CONTINUED)

Field	Value							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	8	8	8	8	8	8	8	8
LED_STEP	10	10	10	10	10	10	10	10

FIGURE 22-5: LINEAR BRIGHTNESS CURVE EXAMPLE



22.9.3.2 Non-linear LED brightness change

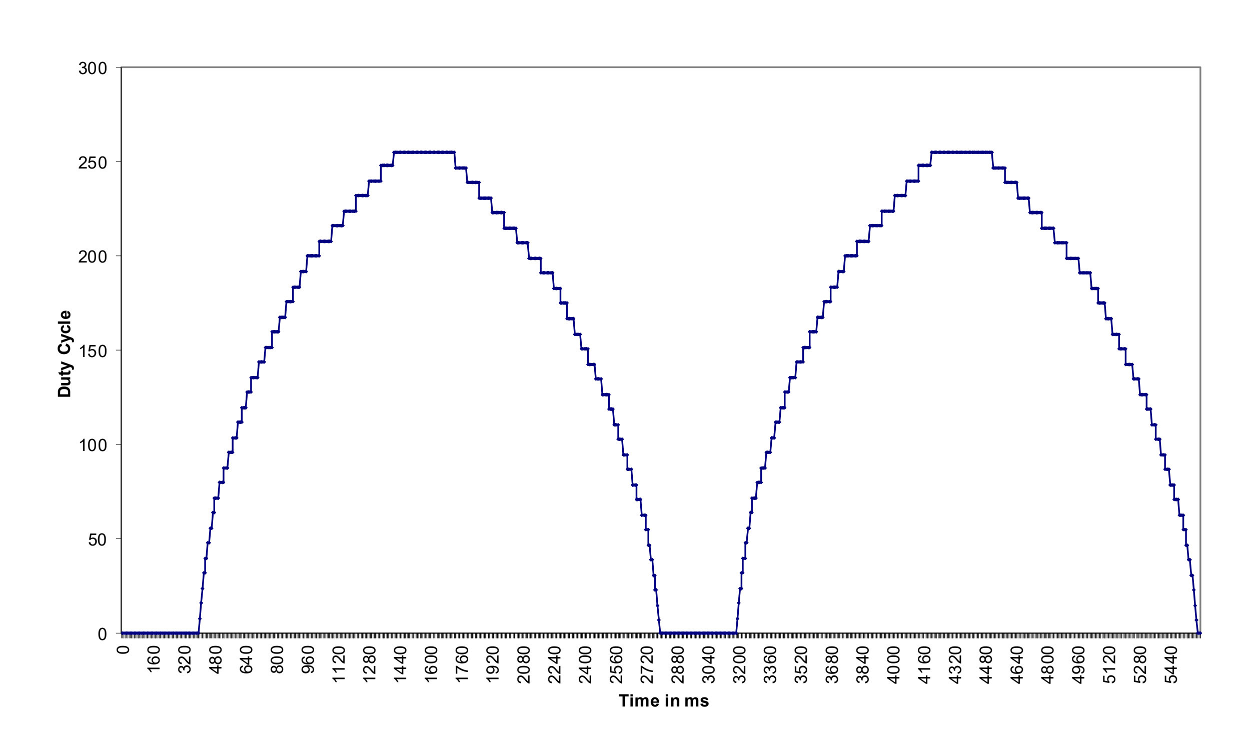
In this example, the brightness of the LED increases and diminishes in a non-linear fashion. The brightness forms a curve that is approximated by four piece wise-linear line segments. The entire cycle takes about 2.8 seconds. The rise time and fall time are about 1 second, with a hold time at maximum brightness of 320ms and a hold time at minimum brightness of 400ms. The LED brightness varies between full off and full on. The PWM size is set to 7-bit, so the time unit for adjusting the PWM is approximately 4ms. The registers are configured as follows:

TABLE 22-9: NON-LINEAR EXAMPLE CONFIGURATION

Field	Value							
PSIZE	7-bit							
MAX	255 (effectively 127)							
MIN	0							
HD	80 ticks (320ms)							
LD	100 ticks (400ms)							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	2	3	6	6	9	9	16	16
LED_STEP	4	4	4	4	4	4	4	4

The resulting curve is shown in the following figure:

FIGURE 22-6: NON-LINEAR BRIGHTNESS CURVE EXAMPLE



22.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Blinking/Breathing LED Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 22-10: REGISTER SUMMARY

Offset	Register Name
00h	LED Configuration Register
04h	LED Limits Register
08h	LED Delay Register
0Ch	LED Update Stepsize Register
10h	LED Update Interval Register
14h	LED Output Delay

In the following register definitions, a “PWM period” is defined by time the PWM counter goes from 000h to its maximum value (FFh in 8-bit mode, FEh in 7-bit mode and FCh in 6-bit mode, as defined by the PSCALE field in register LED_CFG). The end of a PWM period occurs when the PWM counter wraps from its maximum value to 0.

The registers in this block can be written 32-bits, 16-bits or 8-bits at a time. Writes to LED Configuration Register take effect immediately. Writes to LED Limits Register are held in a holding register and only take effect only at the end of a PWM period. The update takes place at the end of every period, even if only one byte of the register was updated. This means that in blink/PWM mode, software can change the duty cycle with a single 8-bit write to the MIN field in the LED_LIMIT register. Writes to LED Delay Register, LED Update Stepsize Register and LED Update Interval Register also go initially into a holding register. The holding registers are copied to the operating registers at the end of a PWM period only if the Enable Update bit in the LED Configuration Register is set to 1. If LED_CFG is 0, data in the holding registers is retained but not copied to the operating registers when the PWM period expires. To change an LED breathing configuration, software should write these three registers with the desired values and then set LED_CFG to 1. This mechanism ensures that all parameters affecting LED breathing will be updated consistently, even if the registers are only written 8 bits at a time.

22.10.1 LED CONFIGURATION REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
16	SYMMETRY 1=The rising and falling ramp times are in Asymmetric mode. Table 22-7, "Asymmetric Breathing Mode Register Usage" shows the application of the Stepsize and Interval registers to the four segments of rising duty cycles and the four segments of falling duty cycles. 0=The rising and falling ramp times (as shown in Figure 22-2, "Breathing LED Example") are in Symmetric mode. Table 22-6, "Symmetric Breathing Mode Register Usage" shows the application of the Stepsize and Interval registers to the 8 segments of both rising and falling duty cycles.	R/W	0b	RESET_SYS
15:8	WDT_RELOAD The PWM Watchdog Timer counter reload value. On system reset, it defaults to 14h, which corresponds to a 4 second Watchdog timeout value.	R/W	14h	RESET_SYS
7	RESET Writes of'1' to this bit resets the PWM registers to their default values. This bit is self clearing. Writes of '0' to this bit have no effect.	W	0b	RESET_SYS
6	ENABLE_UPDATE This bit is set to 1 when written with a '1'. Writes of '0' have no effect. Hardware clears this bit to 0 when the breathing configuration registers are updated at the end of a PWM period. The current state of the bit is readable any time. This bit is used to enable consistent configuration of LED_DELAY, LED_STEP and LED_INT. As long as this bit is 0, data written to those three registers is retained in a holding register. When this bit is 1, data in the holding register are copied to the operating registers at the end of a PWM period. When the copy completes, hardware clears this bit to 0.	R/WS	0b	RESET_SYS
5:4	PWM_SIZE This bit controls the behavior of PWM: 3=Reserved 2=PWM is configured as a 6-bit PWM 1=PWM is configured as a 7-bit PWM 0=PWM is configured as an 8-bit PWM	R/W	0b	RESET_SYS

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	SYNCHRONIZE When this bit is ‘1’, all counters for all LEDs are reset to their initial values. When this bit is ‘0’ in the LED Configuration Register for all LEDs, then all counters for LEDs that are configured to blink or breathe will increment or decrement, as required. To synchronize blinking or breathing, the SYNCHRONIZE bit should be set for at least one LED, the control registers for each LED should be set to their required values, then the SYNCHRONIZE bits should all be cleared. If the all LEDs are set for the same blink period, they will all be synchronized.	R/W	0b	RESET_SYS
2	CLOCK_SOURCE This bit controls the base clock for the PWM. It is only valid when CNTRL is set to blink (2). 1=Clock source is the Main system clock 0=Clock source is the 32.768 KHz clock	R/W	0b	RESET_SYS
1:0	CONTROL This bit controls the behavior of PWM: 3=PWM is always on 2=LED blinking (standard PWM) 1=LED breathing configuration 0=PWM is always off. All internal registers and counters are reset to 0. Clocks are gated	R/W	00b	RESET_SYS
			11b	WDT TC

22.10.2 LED LIMITS REGISTER

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period. The two byte fields may be written independently. Reads of this register return the current contents and not the value of the holding register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:8	MAXIMUM In breathing mode, when the current duty cycle is greater than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field HD in register LED_DELAY, then starts decrementing the current duty cycle	R/W	0h	RESET_SYS
7:0	MINIMUM In breathing mode, when the current duty cycle is less than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field LD in register LED_DELAY, then starts incrementing the current duty cycle In blinking mode, this field defines the duty cycle of the blink function.	R/W	0h	RESET_SYS

22.10.3 LED DELAY REGISTER

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:12	HIGH_DELAY In breathing mode, the number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MAX in register LED_LIMIT. 4095=The current duty cycle is decremented after 4096 PWM periods … 1=The delay counter is bypassed and the current duty cycle is decremented after two PWM period 0=The delay counter is bypassed and the current duty cycle is decremented after one PWM period	R/W	000h	RESET_ SYS
11:0	LOW_DELAY The number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MIN in register LED_LIMIT. 4095=The current duty cycle is incremented after 4096 PWM periods … 0=The delay counter is bypassed and the current duty cycle is incremented after one PWM period In blinking mode, this field defines the prescalar for the PWM clock	R/W	000h	RESET_ SYS

22.10.4 LED UPDATE STEPSIZE REGISTER

This register has eight segment fields which provide the amount the current duty cycle is adjusted at the end of every PWM period. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the SYMMETRY bit in the LED Configuration Register Register)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 22-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

In 8-bit mode, each 4-bit STEPSIZE field represents 16 possible duty cycle modifications, from 1 to 16 as the duty cycle is modified between 0 and 255:

15: Modify the duty cycle by 16

...

1: Modify the duty cycle by 2

0=Modify the duty cycle by 1

In 7-bit mode, the least significant bit of the 4-bit field is ignored, so each field represents 8 possible duty cycle modifications, from 1 to 8, as the duty cycle is modified between 0 and 127:

14, 15: Modify the duty cycle by 8

...

2, 3: Modify the duty cycle by 2

0, 1: Modify the duty cycle by 1

In 6-bit mode, the two least significant bits of the 4-bit field is ignored, so each field represents 4 possible duty cycle modifications, from 1 to 4 as the duty cycle is modified between 0 and 63:

12, 13, 14, 15: Modify the duty cycle by 4

8, 9, 10, 11: Modify the duty cycle by 3

4, 5, 6, 7: Modify the duty cycle by 2

0, 1, 2, 3: Modify the duty cycle by 1

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	UPDATE_STEP7 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 111.	R/W	0h	RESET_SYS
27:24	UPDATE_STEP6 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 110.	R/W	0h	RESET_SYS
23:20	UPDATE_STEP5 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 101	R/W	0h	RESET_SYS
19:16	UPDATE_STEP4 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 100.	R/W	0h	RESET_SYS
15:12	UPDATE_STEP3 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 011.	R/W	0h	RESET_SYS
11:8	UPDATE_STEP2 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 010.	R/W	0h	RESET_SYS
7:4	UPDATE_STEP1 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 001.	R/W	0h	RESET_SYS
3:0	UPDATE_STEP0 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 000.	R/W	0h	RESET_SYS

22.10.5 LED UPDATE INTERVAL REGISTER

This register has eight segment fields which provide the number of PWM periods between updates to current duty cycle. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the SYMMETRY bit in the LED Configuration Register Register)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 22-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	UPDATE_INTERVAL7 The number of PWM periods between updates to current duty cycle when the segment index is equal to 111b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
27:24	UPDATE_INTERVAL6 The number of PWM periods between updates to current duty cycle when the segment index is equal to 110b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
23:20	UPDATE_INTERVAL5 The number of PWM periods between updates to current duty cycle when the segment index is equal to 101b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
19:16	UPDATE_INTERVAL4 The number of PWM periods between updates to current duty cycle when the segment index is equal to 100b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
15:12	UPDATE_INTERVAL3 The number of PWM periods between updates to current duty cycle when the segment index is equal to 011b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
11:8	UPDATE_INTERVAL2 The number of PWM periods between updates to current duty cycle when the segment index is equal to 010b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	UPDATE_INTERVAL1 The number of PWM periods between updates to current duty cycle when the segment index is equal to 001b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_ SYS
3:0	UPDATE_INTERVAL0 The number of PWM periods between updates to current duty cycle when the segment index is equal to 000b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_ SYS

22.10.6 LED OUTPUT DELAY

This register permits the transitions for multiple blinking/breathing LED outputs to be skewed, so as not to present too great a current load. The register defines a count for the number of clocks the circuitry waits before turning on the output, either on initial enable, after a resume from Sleep, or when multiple outputs are synchronized through the Sync control in the LED CONFIGURATION (LED_CFG) register.

When more than one LED outputs are used simultaneously, the LED OUTPUT DELAY fields of each should be configured with different values so that the outputs are skewed. When used with the 32KHz clock domain as a clock source, the differences can be as small as 1.

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	OUTPUT_DELAY The delay, in counts of the clock defined in Clock Source (CLKSRC), in which output transitions are delayed. When this field is 0, there is no added transition delay. When the LED is programmed to be Always On or Always Off, the Output Delay field has no effect.	R/W	000h	RESET_ SYS

23.0 I2C/SMBUS INTERFACE

23.1 Introduction

This section describes the Power Domain, Resets, Clocks, Interrupts, Registers and the Physical Interface of the I2C/SMBus interface. In I2C mode, this block supports Promiscuous mode when configured as I2C slave. For a General Description, Features, Block Diagram, Functional Description, Registers Interface and other core-specific details*, see* Ref [1] (note: in this chapter, *italicized text* typically refers to SMB-I2C Controller core interface elements as described in Ref [1]).

23.2 References

1. I2C_SMB Controller Core with Network Layer Support (SMB2) - 16MHz I2C Baud Clock“, Revision 3.6, Core-Level Architecture Specification, Microchip

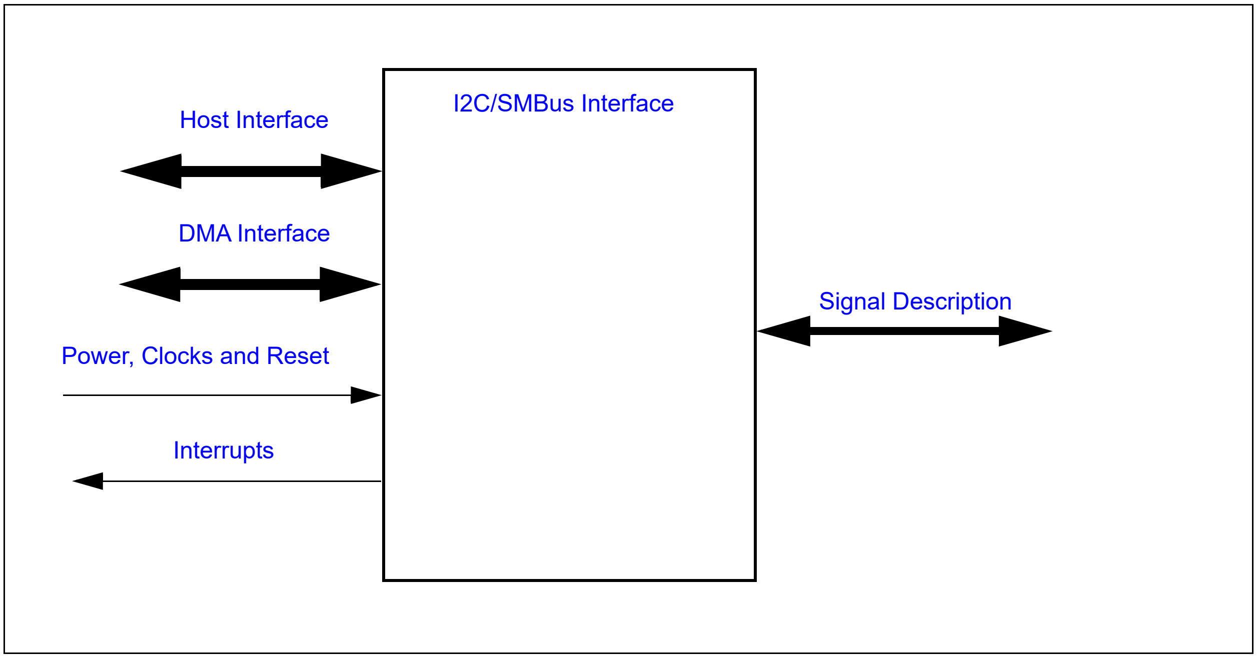
23.3 Terminology

There is no terminology defined for this chapter.

23.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface. In addition, this block is equipped with:

FIGURE 23-1: I/O DIAGRAM OF BLOCK



23.5 Signal Description

see the Pin Configuration section for a description of the SMB-I2C pin configuration.

23.6 Host Interface

The registers defined for the I2C/SMBus Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

23.7 DMA Interface

This block is designed to communicate with the Internal DMA Controller. This feature is defined in the SMB-I2C Controller Core Interface specification (*See* Ref [1]).

Note: For a description of the Internal DMA Controller implemented in this design see Section 7.0, "Internal DMA Controller".

23.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

23.8.1 POWER DOMAINS

Name	Description
VTR_CORE	This power well sources all of the registers and logic in this block, except where noted.

23.8.2 CLOCK INPUTS

Name	Description
16MHz	This is the clock signal drives the SMB-I2C Controller core. The core also uses this clock to generate the SMB-I2C_CLK on the pin interface. It is derived from the main system clock

23.8.3 RESETS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in the SMB-I2C Controller core.

23.9 Interrupts

Source	Description
SMB-I2C	I^2C Activity Interrupt Event
SMB-I2C_WAKE	This interrupt event is triggered when an SMB/I2C Master initiates a transaction by issuing a START bit (a high-to-low transition on the SDA line while the SCL line is high) on the bus currently connected to the SMB-I2C Controller. The EC interrupt handler for this event only needs to clear the interrupt SOURCE bit and return; if the transaction results in an action that requires EC processing, that action will trigger the SMB-I2C interrupt event.

23.10 Low Power Modes

The SMB-I2C Controller may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry.

23.11 Description

23.11.1 SMB-I2C CONTROLLER CORE

The SMB-I2C Controller behavior is defined in the SMB-I2C Controller Core Interface specification (*See* Ref [1]).

23.11.2 PHYSICAL INTERFACE

The Physical Interface for the SMB-I2C Controller core is configurable for up to 10 ports. Each I2C_WAKE Controller can be connected to any of the ports defined in Table 23-1, "SMB-I2C Port Selection". The *PORT SEL [3:0]* bit field in each controller independently sets the port for the controller. The default for each field is Fh, Reserved, which means that the SMB-I2C Controller is not connected to a port.

An I^2C port should be connected to a single controller. An attempt to configure the *PORT SEL [3:0]* bits in one controller to a value already assigned to another controller may result in unexpected results.

The port signal-function names and pin numbers are defined in Pin Configuration section. The I^2C port selection is made using the *PORT SEL [3:0]* bits in the *Configuration Register* as described in Ref [1].. In the Pin section, the SDA (Data) pins are listed asi2Cxx_SDA and the SCL (Clock) pins are listed as I2C*xx*_SCL, where *xx* represents the port number 00 through 15. The CPU-voltage-level port SB_TSI is also listed in the pin section with the SD-TSI_DAT and SD-TSI_CLK.

For I^2C port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as an I^2C port.

For more information regarding the SMB-I2C Controller core see *Section 2.2,* “*Physical Interface*” in Ref[1].

TABLE 23-1: SMB-I2C PORT SELECTION

PORT_SEL[3:0]				**Port**
3	**2**	**1**	**0**	
0	0	0	0	I2C00
0	0	0	1	I2C01
0	0	1	0	I2C02
0	0	1	1	I2C03
0	1	0	0	I2C04
0	1	0	1	I2C05
0	1	1	0	I2C06
0	1	1	1	I2C07
1	0	0	0	I2C08
1	0	0	1	I2C09
1	0	1	0	I2C10
1	0	1	1	I2C11
1	1	0	0	I2C12
1	1	0	1	I2C13
1	1	1	0	I2C14
1	1	1	1	I2C15

Note: Refer toSection 2.4.11for the pin mapping

23.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the SMB-I2C Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Registers for the SMB-I2C Controllers are listed in Reference[1].

24.0 QUAD SPI MASTER CONTROLLER

24.1 Overview

The Quad SPI Master Controller may be used to communicate with various peripheral devices that use a Serial Peripheral Interface, such as EEPROMS, DACs and ADCs. The controller can be configured to support advanced SPI Flash devices with multi-phase access protocols. Data can be transfered in Half Duplex, Single Data Rate, Dual Data Rate and Quad Data Rate modes. In all modes and all SPI clock speeds, the controller supports back-to-back reads and writes without clock stretching if internal bandwidth permits.

24.2 References

No references have been cited for this feature.

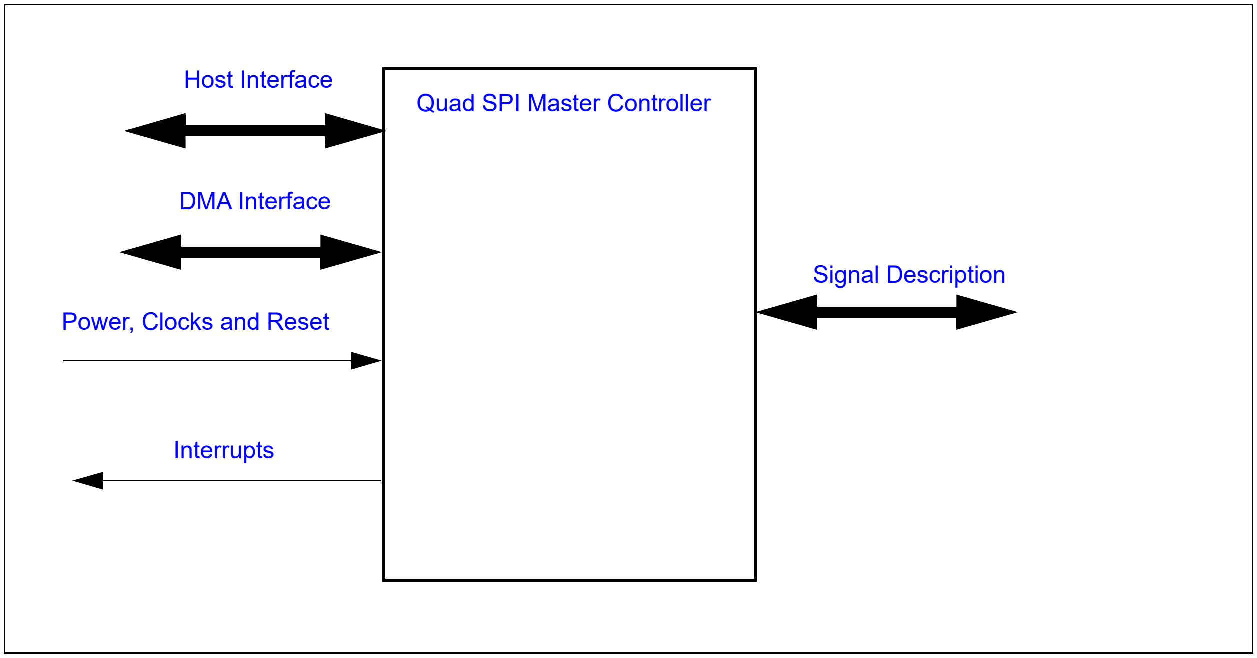
24.3 Terminology

No terminology for this block.

24.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 24-1: I/O DIAGRAM OF BLOCK



24.5 Signal Description

TABLE 24-1: EXTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
SPI_CLK	Output	SPI Clock output used to drive the SPCLK pin.
SPI_CS#	Output	SPI chip select.
SPI_IO0	Input/Output	SPI Data pin 0. Also used as SPI_MOSI, Master-Out/Slave-In when the interface is used in Single wire mode
SPI_IO1	Input/Output	SPI Data pin 1. Also used as SPI_MISO, Master-In/Slave-Out when the interface is used in Single wire mode

TABLE 24-1: EXTERNAL SIGNAL DESCRIPTION (CONTINUED)

Name	Direction	Description
SPI_IO2	Input/Output	SPI Data pin 2 when the SPI interface is used in Quad Mode. Also can be used by firmware as WP.
SPI_IO3	Input/Output	SPI Data pin 3 when the SPI interface is used in Quad Mode. Also can be used by firmware as HOLD.

24.6 Host Interface

The registers defined for the General Purpose Serial Peripheral Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

24.7 DMA Interface

This block is designed to communicate with the Internal DMA Controller.

Note: For a description of the Internal DMA Controller implemented in this design see Section 7.0, "Internal DMA Controller".

24.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

24.8.1 POWER

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

24.8.2 CLOCKS

Name	Description
48MHz	This is a clock source for the SPI clock generator.

24.8.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.QMSPI Status Register
RESET	This reset is generated if either the RESET_SYS is asserted or the SOFT_RESET is asserted.

24.9 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
QMSPI_INT	Interrupt generated by the Quad SPI Master Controller. Events that may cause the interrupt to be asserted are stored in the QMSPI Status Register.

24.10 Low Power Modes

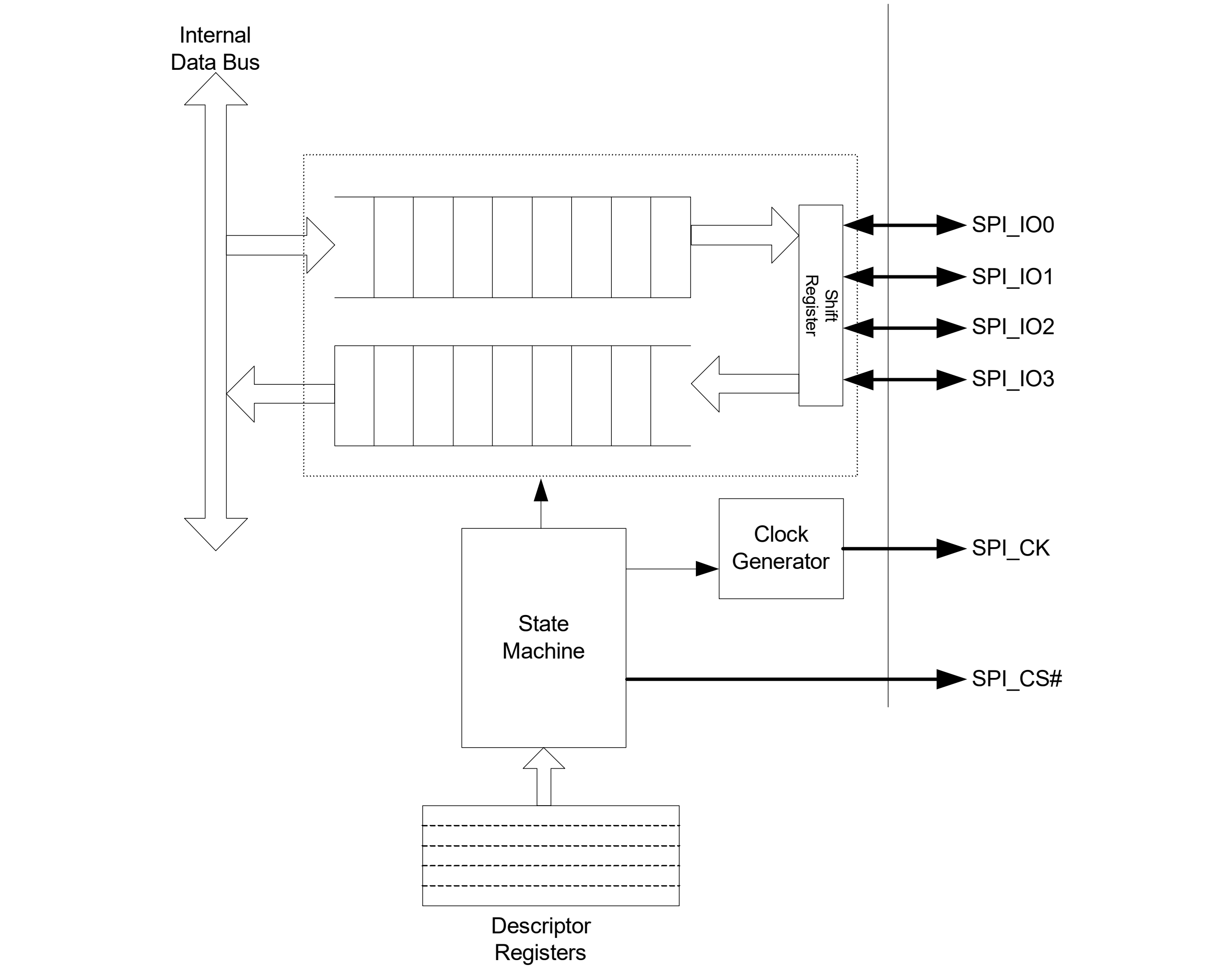
The Quad SPI Master Controller is always in its lowest power state unless a transaction is in process. A transaction is in process between the time the START bit is written with a ‘1’ and the TRANSFER_DONE bit is set by hardware to ‘1’.

If the QMSPI SLEEP_ENABLE input is asserted, writes to the START bit are ignored and the Quad SPI Master Controller will remain in its lowest power state.

24.11 Description

- Support for multiple SPI pin configurations
  - Single wire half duplex
  - Two wire full duplex
  - Two wire double data rate
  - Four wire quad data rate
- Separate FIFO buffers for Receive and Transmit
  - 8 byte FIFO depth in each FIFO
  - Each FIFO can be 1 byte, 2 bytes or 4 bytes wide
- Support for all four SPI clock formats
- Programmable SPI Clock generator, with clock polarity and phase controls
- Separate DMA support for Receive and Transmit data transfers
- Configurable interrupts, for errors, individual bytes, or entire transactions
- Descriptor Mode, in which a set of sixteen descriptor registers can configure the controller to autonomously perform multi-phase SPI data transfers
- Capable of wire speed transfers in all SPI modes and all configurable SPI clock rates (internal bus contention may cause clock stretching)

FIGURE 24-2: QUAD MASTER SPI BLOCK DIAGRAM



24.11.1 SPI CONFIGURATIONS MODES

- Half Duplex. All SPI data transfers take place on a single wire, SPI_IO0
- Full Duplex. This is the legacy SPI configuration, where all SPI data is transfered one bit at a time and data from the SPI Master to the SPI Slave takes place on SPI_MOSI (SPI_IO0) and at the same time data from the SPI Slave to the SPI Master takes place on SPI_MISO (SPI_IO1)
- Dual Data Rate. Data transfers between the SPI Master and the SPI Slave take place two bits at a time, using SPI_IO0 and SPI_IO1
- Quad Data Rate. Data transfers between the SPI Master and the SPI Slave take place four bits at a time, using all four SPI data wires, SPI_IO0, SPI_IO1, SPI_IO2 and SPI_IO3

24.11.2 SPI CONTROLLER MODES

- Manual. In this mode, firmware control all SPI data transfers byte at a time
- DMA. Firmware configures the SPI Master controller for characteristics like data width but the transfer of data between the FIFO buffers in the SPI controller and memory is controlled by the DMA controller. DMA transfers can take place from the Slave to the Master, from the Master to the Slave, or in both directions simultaneously
- Descriptor. Descriptor Mode extends the SPI Controller so that firmware can configure a multi-phase SPI transfer, in which each phase may have a different SPI bus width, a different direction, and a different length. For example, firmware can configure the controller so that a read from an advanced SPI Flash, which consists of a command phase, an address phase, a dummy cycle phase and the read phase, can take place as a single operation, with a single interrupt to firmware when the entire transfer is completed

24.11.3 SPI CLOCK

The SPI output clock is derived from the 48MHz, divided by a value programmed in the CLOCK_DIVIDE field of the QMSPI Mode Register. Sample frequencies are shown in the following table:

TABLE 24-2: EXAMPLE SPI FREQUENCIES

CLOCK_DIVIDE	SPI Clock Frequency
0	187.5 KHz
1	48 MHz
2	24 MHz
3	16 MHz
6	8 MHz
48	1 MHz
128	375 KHz
255	188.25 KHz

24.11.4 ERROR CONDITIONS

The Quad SPI Master Controller can detect some illegal configurations. When these errors are detected, an error is signaled via the PROGRAMMING_ERROR status bit. This bit is asserted when any of the following errors are detected:

- Both Receive and the Transmit transfers are enabled when the SPI Master Controller is configured for Dual Data Rate or Quad Data Rate
- Both Pull-up and Pull-down resistors are enabled on either the Receive data pins or the Transmit data pins
- The transfer length is programmed in bit mode, but the total number of bits is not a multiple of 2 (when the controller is configured for Dual Data Rate) or 4 (when the controller is configured for Quad Data Rate)
- Both the STOP bit and the START bits in the QMSPI Execute Register are set to ‘1’ simultaneously

24.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Quad SPI Master Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 24-3: REGISTER SUMMARY

Offset	Register Name
0h	QMSPI Mode Register
4h	QMSPI Control Register
8h	QMSPI Execute Register
Ch	QMSPI Interface Control Register
10h	QMSPI Status Register
14h	QMSPI Buffer Count Status Register
18h	QMSPI Interrupt Enable Register
1Ch	QMSPI Buffer Count Trigger Register
20h	QMSPI Transmit Buffer Register
24h	QMSPI Receive Buffer Register
28h	QMSPI Chip Select Timing Register
30h	QMSPI Description Buffer 0 Register
34h	QMSPI Description Buffer 1 Register
38h	QMSPI Description Buffer 2 Register
3Ch	QMSPI Description Buffer 3 Register
40h	QMSPI Description Buffer 4 Register
44h	QMSPI Description Buffer 5 Register
48h	QMSPI Description Buffer 6 Register
4Ch	QMSPI Description Buffer 7 Register
50h	QMSPI Description Buffer 8 Register
54h	QMSPI Description Buffer 9 Register
58h	QMSPI Description Buffer 10 Register
5Ch	QMSPI Description Buffer 11 Register
60h	QMSPI Description Buffer 12 Register
64h	QMSPI Description Buffer 13 Register
68h	QMSPI Description Buffer 14 Register
6Ch	QMSPI Description Buffer 15 Register
B0	Test

24.12.1 QMSPI MODE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:16	CLOCK_DIVIDE The SPI clock divide in number of system clocks. A value of 1 divides the master clock by 1, a value of 255 divides the master clock by 255. A value of 0 divides the master clock by 256. See Table 24-2, "Example SPI Frequencies" for examples.	R/W	0h	RESET
15:14	Reserved	RES	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
13:12	CHIP_SELECT Selects which Chip Select line is active. The non-active CS line is driven high. 00=Chip Select 0 01=Chip Select 1 1x=unused.	R/W	0h	RESET
11	Reserved	RES	-	-
10	CHPA_MISO If CPOL=1: 1=Data are captured on the rising edge of the SPI clock 0=Data are captured on the falling edge of the SPI clock If CPOL=0: 1=Data are captured on the falling edge of the SPI clock 0=Data are captured on the rising edge of the SPI clock Application Notes: Common SPI Mode configurations: Common SPI Modes require the CHPA_MISO and CHPA_MOSI programmed to the same value. E.g., - Mode 0: CPOL=0; CHPA_MISO=0; CHPA_MOSI=0 - Mode 3: CPOL=1; CHPA_MISO=1; CHPA_MOSI=1 Alternative SPI Mode configurations When configured for quad mode, applications operating at 48MHz may find it difficult to meet the minimum setup timing using the default Mode 0. It is recommended to configure the Master to sample and change data on the same edge when operating at 48MHz as shown in these examples. E.g, - Mode 0: CPOL=0; CHPA_MISO=1; CHPA_MOSI=0 - Mode 3: CPOL=1; CHPA_MISO=0; CHPA_MOSI=1	R/W	0h	RESET
9	CHPA_MOSI If CPOL=1: 1=Data changes on the falling edge of the SPI clock 0=Data changes on the rising edge of the SPI clock If CPOL=0: 1=Data changes on the rising edge of the SPI clock 0=Data changes on the falling edge of the SPI clock	R/W	0h	RESET
8	CPOL Polarity of the SPI clock line when there are no transactions in process. 1=SPI Clock starts High 0=SPI Clock starts Low	R/W	0h	RESET
7:3	Reserved	RES	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	SAF DMA Mode This mode enables the H/W to allow a DMA to access the part with accesses that are not a multiple of 4 bytes. 0 = Standard DMA functionality 1 = SAF DMA Mode: Non-standard DMA functionality with arbitrary (unaligned) sizes and FIFO underflow allowed.	R/W	0h	RESET
1	SOFT_RESET Writing this bit with a ‘1’ will reset the Quad SPI block. It is self-clearing.	W	0h	RESET_SYS
0	ACTIVATE 1=Enabled. The block is fully operational 0=Disabled. Clocks are gated to conserve power and the output signals are set to their inactive state	R/W	0h	RESET

24.12.2 QMSPI CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	TRANSFER_LENGTH The length of the SPI transfer. The count is in bytes or bits, depending on the value of TRANSFER_UNITS. A value of '0' means an infinite length transfer.	R/W	0h	RESET
16	DESCRIPTION_BUFFER_ENABLE This enables the Description Buffers to be used. 1=Description Buffers in use. The first buffer is defined in DESCRIPTION_BUFFER_POINTER 0=Description Buffers disabled	R/W	0h	RESET
15:12	DESCRIPTION_BUFFER_POINTER This field selects the first buffer used if Description Buffers are enabled.	R/W	0h	RESET
11:10	TRANSFER_UNITS 3=TRANSFER_LENGTH defined in units of 16-byte segments 2=TRANSFER_LENGTH defined in units of 4-byte segments 1=TRANSFER_LENGTH defined in units of bytes 0=TRANSFER_LENGTH defined in units of bits	R/W	0h	RESET
9	CLOSE_TRANSFER_ENABLE This selects what action is taken at the end of a transfer. When the transaction closes, the Chip Select de-asserts, the SPI interface returns to IDLE and the DMA transfer terminates. When Description Buffers are in use this bit must be set only on the Last Buffer. 1=The transaction is terminated 0=The transaction is not terminated	R/W	0h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5:4	TX_DMA_ENABLE This bit enables DMA support for Transmit Transfer. If enabled, DMA will be requested to fill the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Transmit Buffer must be emptied by firmware	R/W	0h	RESET
3:2	TX_TRANSFER_ENABLE This field bit selects the transmit function of the SPI interface. 3=Transmit Enabled in 1 Mode. The MOSI or IO Bus will send out only 1's. The Transmit Buffer will not be used 2=Transmit Enabled in 0 Mode. The MOSI or IO Bus will send out only 0's. The Transmit Buffer will not be used. 1=Transmit Enabled. Data will be fetched from the Transmit Buffer and sent out on the MOSI or IO Bus. 0=Transmit is Disabled. Not data is sent. This will cause the MOSI be to be undriven, or the IO bus to be undriven if Receive is also disabled.	R/W	0h	RESET
1:0	INTERFACE_MODE This field sets the transmission mode. If this field is set for Dual Mode or Quad Mode then either TX_TRANSFER_ENABLE or RX_TRANSFER_ENABLE **must** be 0. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single/Duplex Mode	R/W	0h	RESET

24.12.3 QMSPI EXECUTE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2	CLEAR_DATA_BUFFER Writing a ‘1’ to this bit will clear out the Transmit and Receive FIFOs. Any data stored in the FIFOs is discarded and all count fields are reset. Writing a ‘0’ to this bit has no effect. This bit is self-clearing.	W	0h	RESET
1	STOP Writing a ‘1’ to this bit will stop any transfer in progress at the next byte boundary. Writing a ‘0’ to this bit has no effect. This bit is self-clearing. This bit must not be set to ‘1’ if the field START in this register is set to ‘1’.	W	0h	RESET

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	START Writing a ‘1’ to this bit will start the SPI transfer. Writing a ‘0’ to this bit has no effect. This bit is self-clearing. This bit must not be set to ‘1’ if the field STOP in this register is set to ‘1’.	W	0h	RESET

24.12.4 QMSPI INTERFACE CONTROL REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	PULLUP_ON_NOT_DRIVEN 1=Enable pull-up resistors on Transmit pins while the pins are not driven 0=No pull-up resistors enabled ion Transmit pins	R/W	0h	RESET
6	PULLDOWN_ON_NOT_DRIVEN 1=Enable pull-down resistors on Transmit pins while the pins are not driven 0=No pull-down resistors enabled ion Transmit pins	R/W	0h	RESET
5	PULLUP_ON_NOT_SELECTED 1=Enable pull-up resistors on Receive pins while the SPI Chip Select signal is not asserted 0=No pull-up resistors enabled on Receive pins	R/W	0h	RESET
4	PULLDOWN_ON_NOT_SELECTED 1=Enable pull-down resistors on Receive pins while the SPI Chip Select signal is not asserted 0=No pull-down resistors enabled on Receive pins	R/W	0h	RESET
3	HOLD_OUT_ENABLE 1=HOLD SPI Output Port is driven 0=HOLD SPI Output Port is not driven	R/W	0h	RESET
2	HOLD_OUT_VALUE This bit sets the value on the HOLD SPI Output Port if it is driven. 1=HOLD is driven to 1 0=HOLD is driven to 0	R/W	0h	RESET
1	WRITE_PROTECT_OUT_ENABLE 1=WRITE PROTECT SPI Output Port is driven 0=WRITE PROTECT SPI Output Port is not driven	R/W	0h	RESET

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	WRITE_PROTECT_OUT_VALUE This bit sets the value on the WRITE PROTECT SPI Output Port if it is driven. 1=WRITE PROTECT is driven to 1 0=WRITE PROTECT is driven to 0	R/W	0h	RESET

24.12.5 QMSPI STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	RES	-	-
27:24	CURRENT_DESCRIPTION_BUFFER This field shows the Description Buffer currently active. This field has no meaning if Description Buffers are not enabled.	R	0h	RESET
23:17	Reserved	RES	-	-
16	TRANSFER_ACTIVE 1=A transfer is currently executing 0=No transfer currently in progress	R	0h	RESET
15	RECEIVE_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to write to a full Receive Buffer) 0=No stalls occurred	R/WC	0h	RESET
14	RECEIVE_BUFFER_REQUEST This status is asserted if the Receive Buffer reaches a high water mark established by the RECEIVE_BUFFER_TRIGGER field. 1=RECEIVE_BUFFER_COUNT is greater than or equal to RECEIVE_BUFFER_TRIGGER 0=RECEIVE_BUFFER_COUNT is less than RECEIVE_BUFFER_TRIGGER	R/WC	0h	RESET
13	RECEIVE_BUFFER_EMPTY 1=The Receive Buffer is empty 0=The Receive Buffer is not empty	R	1h	RESET
12	RECEIVE_BUFFER_FULL 1=The Receive Buffer is full 0=The Receive Buffer is not full	R	0h	RESET
11	TRANSMIT_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to read from an empty Transmit Buffer) 0=No stalls occurred	R/WC	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TRANSMIT_BUFFER_REQUEST This status is asserted if the Transmit Buffer reaches a high water mark established by the TRANSMIT_BUFFER_TRIGGER field. 1=TRANSMIT_BUFFER_COUNT is less than or equal to TRANSMIT_BUFFER_TRIGGER 0=TRANSMIT_BUFFER_COUNT is greater than TRANSMIT_BUFFER_TRIGGER	R/WC	0h	RESET
9	TRANSMIT_BUFFER_EMPTY 1=The Transmit Buffer is empty 0=The Transmit Buffer is not empty	R	1h	RESET
8	TRANSMIT_BUFFER_FULL 1=The Transmit Buffer is full 0=The Transmit Buffer is not full	R	0h	RESET
7:5	Reserved	RES	-	-
4	PROGRAMMING_ERROR This bit if a programming error is detected. Programming errors are listed in Section 24.11.4, "Error Conditions". 1=Programming Error detected 0=No programming error detected	R/WC	0h	RESET
3	RECEIVE_BUFFER_ERROR 1=Underflow error occurred (attempt to read from an empty Receive Buffer) 0=No underflow occurred	R/WC	0h	RESET
2	TRANSMIT_BUFFER_ERROR 1=Overflow error occurred (attempt to write to a full Transmit Buffer) 0=No overflow occurred	R/WC	0h	RESET
1	DMA_COMPLETE This field has no meaning if DMA is not enabled. This bit will be set to ‘1’ when the DMA controller asserts the DONE signal to the SPI controller. This occurs either when the SPI controller has closed the DMA transfer, or the DMA channel has completed its count. If both Transmit and Receive DMA transfers are active, then this bit will only assert after both have completed. If CLOSE_TRANSFER_ENABLE is enabled, DMA_COMPLETE and TRANSFER_COMPLETE will be asserted simultaneously. This status is not inhibited by the description buffers, so it can fire on all valid description buffers while operating in that mode. 1=DMA completed 0=DMA not completed	R/WC	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	TRANSFER_COMPLETE In Manual Mode (neither DMA nor Description Buffers are enabled), this bit will be set to ‘1’ when the transfer matches TRANSFER_LENGTH. If DMA Mode is enabled, this bit will be set to ‘1’ when DMA_COMPLETE is set to ‘1’. In Description Buffer Mode, this bit will be set to ‘1’ only when the Last Buffer completes its transfer. In all cases, this bit will be set to ‘1’ if the STOP bit is set to ‘1’ and the controller has completed the current 8 bits being copied. 1=Transfer completed 0=Transfer not complete	R/WC	0h	RESET

24.12.6 QMSPI BUFFER COUNT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_COUNT This is a count of the number of bytes currently valid in the Receive Buffer.	R	0h	RESET
15:0	TRANSMIT_BUFFER_COUNT This is a count of the number of bytes currently valid in the Transmit Buffer.	R	0h	RESET

24.12.7 QMSPI INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14	RECEIVE_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
13	RECEIVE_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	1h	RESET
12	RECEIVE_BUFFER_FULL_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
11	Reserved	RES	-	-

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TRANSMIT_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
9	TRANSMIT_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	0h	RESET
8	TRANSMIT_BUFFER_FULL_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
7:5	Reserved	RES	-	-
4	PROGRAMMING_ERROR_ENABLE 1=Enable an interrupt if PROGRAMMING_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
3	RECEIVE_BUFFER_ERROR_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
2	TRANSMIT_BUFFER_ERROR_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
1	DMA_COMPLETE_ENABLE 1=Enable an interrupt if DMA_COMPLETE is asserted 0=Disable the interrupt	R/W	0h	RESET
0	TRANSFER_COMPLETE_ENABLE 1=Enable an interrupt if TRANSFER_COMPLETE is asserted 0=Disable the interrupt	R/W	0h	RESET

24.12.8 QMSPI BUFFER COUNT TRIGGER REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_TRIGGER An interrupt is triggered if the RECEIVE_BUFFER_COUNT field is greater than or equal to this value. A value of ‘0’ disables the interrupt.	R/W	0h	RESET
15:0	TRANSMIT_BUFFER_TRIGGER An interrupt is triggered if the TRANSMIT_BUFFER_COUNT field is less than or equal to this value. A value of ‘0’ disables the interrupt.	R/W	0h	RESET

24.12.9 QMSPI TRANSMIT BUFFER REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	TRANSMIT_BUFFER Writes to this register store data to be transmitted from the SPI Master to the external SPI Slave. Writes to this block will be written to the Transmit FIFO. A 1 Byte write fills 1 byte of the FIFO. A Word write fills 2 Bytes and a Doubleword write fills 4 bytes. The data must always be aligned to the bottom most byte (so 1 byte write is on bits [7:0] and Word write is on [15:0]). An overflow condition,TRANSMIT_BUFFER_ERROR will happen, if a write to a full FIFO occurs. Write accesses to this register increment the TRANSMIT_BUFFER_COUNT field.	W	0h	RESET

24.12.10 QMSPI RECEIVE BUFFER REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	RECEIVE_BUFFER Buffer that stores data from the external SPI Slave device to the SPI Master (this block), which is received over MISO or IO. Reads from this register will empty the Rx FIFO. A 1 Byte read will have valid data on bits [7:0] and a Word read will have data on bits [15:0]. It is possible to request more data than the FIFO has (underflow condition), but this will cause an error (RECEIVE_BUFFER_ERROR). Read accesses to this register decrement the RECEIVE_BUFFER_COUNT field.	R	0h	RESET

24.12.11 QMSPI CHIP SELECT TIMING REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	DELAY_CS_OFF_TO_CS_ON This selects the number of system clock cycles between CS deassertion to CS assertion. This is the minimum pulse width of CS deassertion. **Note:** this field delays the start of the next transaction, it does not delay the status of the current transaction.	R/W	06h	RESET
23:20	Reserved	RES	0h	RESET

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
19:16	DELAY_LAST_DATA_HOLD This selects the number of system clock cycles between CS deassertion to the data ports for WP and HOLD switching from input to output. This is only used if the WP/HOLD functions are in use and only on IO2/WP and IO3/HOLD pins.	R/W	6h	RESET
15:12	Reserved	RES	0h	RESET
11:8	DELAY_CLK_STOP_TO_CS_OFF This selects the number of system clock cycles between the last clock edge and the deassertion of CS.	R/W	4h	RESET
7:4	Reserved	RES	0h	RESET
3:0	DELAY_CS_ON_TO_CLOCK_START This selects the number of system clock cycles between CS assertion to the start of the SPI Clock. An additional ½ SPI Clock delay is inherently added to allow pre-set-up of the data ports.	R/W	6h	RESET

24.12.12 QMSPI DESCRIPTION BUFFER 0 REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	TRANSFER_LENGTH The length of the SPI transfer. The count is in bytes or bits, depending on the value of TRANSFER_LENGTH_BITS. A value of '0' means an infinite length transfer.	R/W	0h	RESET
16	DESCRIPTION_BUFFER_LAST If this bit is '1' then this is the last Description Buffer in the chain. When the transfer described by this buffer completes the TRANSFER_COMPLETE status will be set to '1'. If this bit is '0', then this is not the last buffer in use. When the transfer completes the next buffer will be activated, and no additional status will be asserted.	R/W	0h	RESET
15:12	DESCRIPTION_BUFFER_NEXT_POINTER This defines the next buffer to be used if Description Buffers are enabled and this is not the last buffer. This can point to the current buffer, creating an infinite loop.	R/W	0h	RESET
11:10	TRANSFER_UNITS 3=TRANSFER_LENGTH defined in units of 16-byte segments 2=TRANSFER_LENGTH defined in units of 4-byte segments 1=TRANSFER_LENGTH defined in units of bytes 0=TRANSFER_LENGTH defined in units of bits	R/W	0h	RESET

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9	CLOSE_TRANFSER_ENABLE This selects what action is taken at the end of a transfer. This bit must be set only on the Last Buffer. 1=The transfer is terminated. The Chip Select de-asserts, the SPI interface returns to IDLE and the DMA interface completes the transfer. 0=The transfer is not closed. Chip Select remains asserted and the DMA interface and the SPI interface remain active	R/W	0h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET
5:4	TX_DMA_ENABLE This bit enables DMA support for Transmit Transfer. If enabled, DMA will be requested to fill the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Transmit Buffer must be emptied by firmware	R/W	0h	RESET
3:2	TX_TRANSFER_ENABLE This field bit selects the transmit function of the SPI interface. 3=Transmit Enabled in 1 Mode. The MOSI or IO Bus will send out only 1's. The Transmit Buffer will not be used 2=Transmit Enabled in 0 Mode. The MOSI or IO Bus will send out only 0's. The Transmit Buffer will not be used. 1=Transmit Enabled. Data will be fetched from the Transmit Buffer and sent out on the MOSI or IO Bus. 0=Transmit is Disabled. No data is sent. This will cause the MOSI be to be undriven, or the IO bus to be undriven if Receive is also disabled.	R/W	0h	RESET

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1:0	INTERFACE_MODE This field sets the transmission mode. If this field is set for Dual Mode or Quad Mode then either TX_TRANSFER_ENABLE or RX_TRANSFER_ENABLE **must** be 0. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single/Duplex Mode	R/W	0h	RESET

24.12.13 QMSPI DESCRIPTION BUFFER 1 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.14 QMSPI DESCRIPTION BUFFER 2 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.15 QMSPI DESCRIPTION BUFFER 3 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.16 QMSPI DESCRIPTION BUFFER 4 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.17 QMSPI DESCRIPTION BUFFER 5 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.18 QMSPI DESCRIPTION BUFFER 6 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.19 QMSPI DESCRIPTION BUFFER 7 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.20 QMSPI DESCRIPTION BUFFER 8 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.21 QMSPI DESCRIPTION BUFFER 9 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.22 QMSPI DESCRIPTION BUFFER 10 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.23 QMSPI DESCRIPTION BUFFER 11 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.24 QMSPI DESCRIPTION BUFFER 12 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.25 QMSPI DESCRIPTION BUFFER 13 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.26 QMSPI DESCRIPTION BUFFER 14 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

24.12.27 QMSPI DESCRIPTION BUFFER 15 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

25.0 TRACE FIFO DEBUG PORT (TFDP)

25.1 Introduction

The TFDP serially transmits Embedded Controller (EC)-originated diagnostic vectors to an external debug trace system.

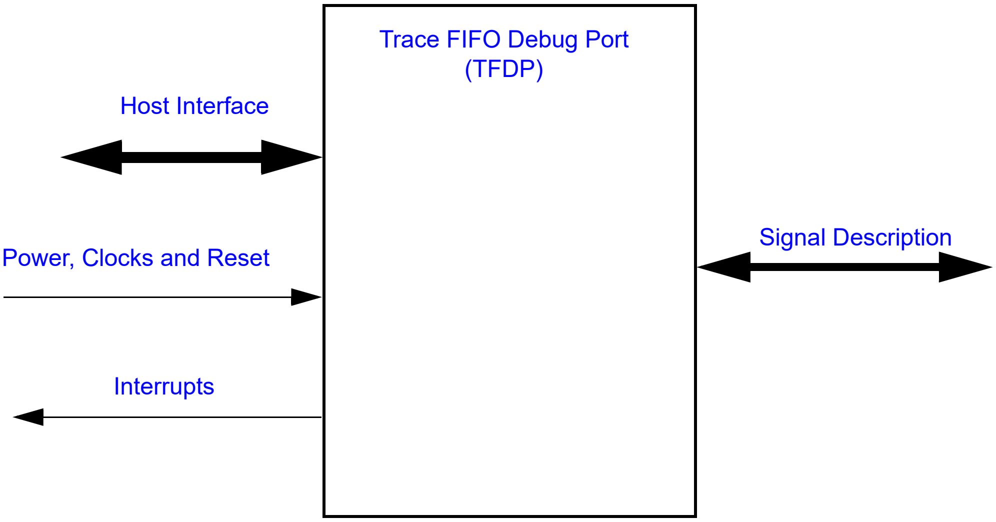
25.2 References

No references have been cited for this chapter.

25.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 25-1: I/O DIAGRAM OF BLOCK



25.4 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 25-1: SIGNAL DESCRIPTION

Name	Direction	Description
TFDP Clk	Output	Derived from EC Bus Clock.
TFDP Data	Output	Serialized data shifted out by TFDP Clk.

25.5 Host Interface

The registers defined for the Trace FIFO Debug Port (TFDP) are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

25.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

25.6.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

25.6.2 CLOCK INPUTS

Name	Description
48MHz	This is the main system clock.

25.6.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

25.7 Interrupts

There are no interrupts generated from this block.

25.8 Low Power Modes

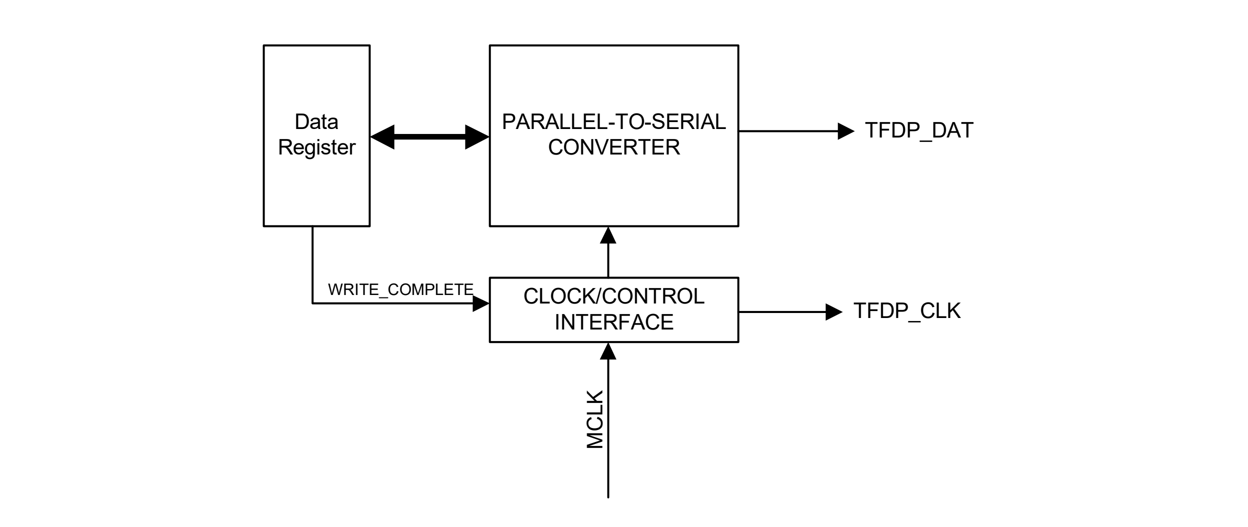
The Trace FIFO Debug Port (TFDP) may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

25.9 Description

The TFDP is a unidirectional (from processor to external world) two-wire serial, byte-oriented debug interface for use by processor firmware to transmit diagnostic information.

The TFDP consists of the Debug Data Register, Debug Control Register, a Parallel-to-Serial Converter, a Clock/Control Interface and a two-pin external interface (TFDP Clk, TFDP Data). See Figure 25-2.

FIGURE 25-2: BLOCK DIAGRAM OF TFDP DEBUG PORT

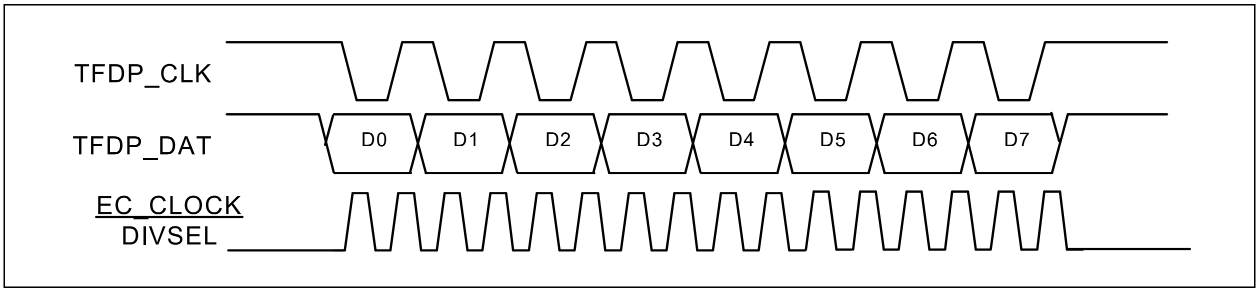


The firmware executing on the embedded controller writes to the Debug Data Register to initiate a transfer cycle (Figure 25-2). The Debug Data Register is loaded into a shift register and shifted out on TFDP_DAT LSB first at the programmed TFDP_CLK Clock rate (Figure 25-3).

Data is transferred in one direction only from the Debug Data Register to the external interface. The data is shifted out at the clock edge. The clock edge is selected by the EDGE_SEL bit in the Debug Control Register. After being shifted out, valid data will be presented at the opposite edge of the TFDP_CLK. For example, when the EDGE_SEL bit is ‘0’ (default), valid data will be presented on the falling edge of the TFDP_CLK. The Setup Time (to the falling edge of TFDP_CLK) is 10 ns, minimum. The Hold Time is 1 ns, minimum.

When the Serial Debug Port is inactive, the TFDP_CLK and TFDP_DAT outputs are ‘1.’ The EC Bus Clock clock input is the transfer clock.

FIGURE 25-3: DATA TRANSFER



25.10 EC-Only Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the Trace FIFO Debug Port (TFDP) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 25-2: REGISTER SUMMARY

Offset	Register Name
00h	Debug Data Register
04h	Debug Control Register

25.10.1 DEBUG DATA REGISTER

The Debut Data Register is Read/Write. It always returns the last data written by the TFDP or the power-on default ‘00h’.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA Debug data to be shifted out on the TFDP Debug port. While data is being shifted out, the Host Interface will ‘hold-off’ additional writes to the data register until the transfer is complete.	R/W	00h	RESET_SYS

25.10.2 DEBUG CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Reserved	RES	-	-
6:4	IP_DELAY Inter-packet Delay. The delay is in terms of TFDP Debug output clocks. A value of 0 provides a 1 clock inter-packet period, while a value of 7 provides 8 clocks between packets:	R/W	000b	RESET_SYS
3:2	DIVSEL Clock Divider Select. The TFDP Debug output clock is determined by this field, according to Table 25-3, "TFDP Debug Clocking":	R/W	00b	RESET_SYS
1	EDGE_SEL 1=Data is shifted out on the falling edge of the debug clock 0=Data is shifted out on the rising edge of the debug clock (Default)	R/W	0b	RESET_SYS
0	EN Enable. 1=Clock enabled 0=Clock is disabled (Default)	R/W	0b	RESET_SYS

TABLE 25-3: TFDP DEBUG CLOCKING

divsel	TFDP Debug Clock
00	24 MHz
01	12 MHz
10	6 MHz
11	Reserved

26.0 VBAT-POWERED CONTROL INTERFACE

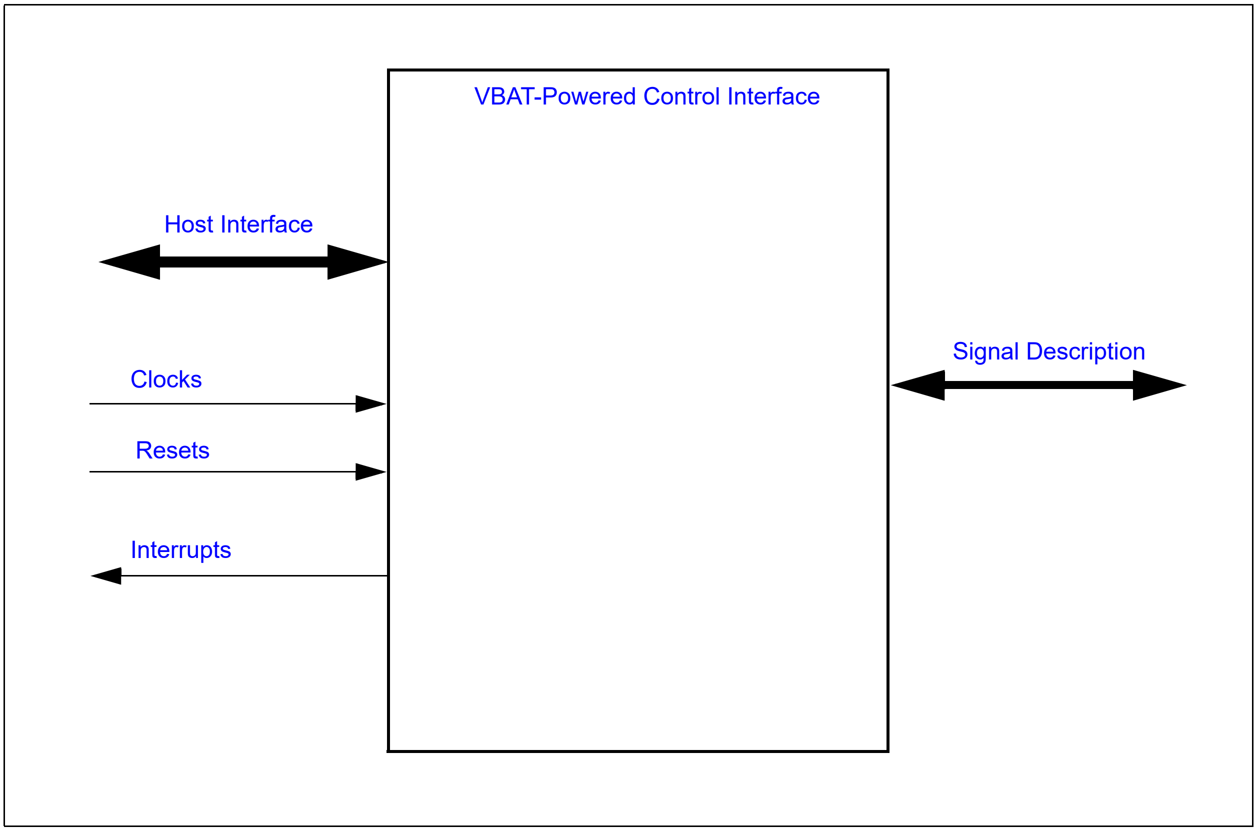
26.1 General Description

The VBAT-Powered Control Interface (VCI) has VBAT-powered combinational logic and input and output signal pins. The block interfaces with the Real Time Clock as well as the Week Alarm.

26.2 Interface

This block is designed to be accesses externally via the pin interface and internally via a registered host interface.

FIGURE 26-1: I/O DIAGRAM OF BLOCK



26.3 Signal Description

TABLE 26-1: EXTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
VCI_IN[3:0]	INPUT	Active-low inputs that can cause wakeup or interrupt events. **Note:** The VCI IP supports up to seven VCI_IN inputs. These inputs are generically referred to as VCI_INx. Input signals not routed to pins or balls on the package are connected to VBAT.
VCI_OUT	OUTPUT	Output status driven by this block.

TABLE 26-2: INTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
Week_Alarm	INPUT	Signal from the Week Timer block. The alarm is asserted by the timer when the Week_Alarm Power-Up Output is asserted
RTC_Alarm	INPUT	Signal from the Real Time Clock block. The alarm is asserted by the RTC when the RTC_ALRM signal is asserted.
VTR_PWRGD	INPUT	Status signal for the state of the VTR power rail. This signal is high if the power rail is on, and low if the power rail is off.

26.4 Host Interface

The registers defined for the VBAT-Powered Control Interface are accessible only by the EC.

26.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

26.5.1 POWER DOMAINS

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR_CORE	The power well sources register access by the host. The block continues to operate internally while this rail is down

26.5.2 CLOCKS

This block does not require clocks.

26.5.3 RESETS

Name	Description
RESET_VBAT	This reset signal is used reset all of the registers and logic in this block.
RESET_SYS	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR_CORE powered circuitry on-chip. Otherwise it has no effect on the internal state.

26.6 Interrupts

Source	Description
VCI_IN[3:0]	These interrupts are routed to the Interrupt Aggregator. They are only asserted when both VBAT and VTR_CORE are powered. Edge detection and assertion level for the interrupt are configured in the GPIO Pin Control Registers for the GPIOs that shares pins with VCI_INx# inputs. The interrupts are equivalent to the GPIO interrupts for the GPIOs that share the pins, but appear on different registers in the Interrupt Aggregator.

26.7 Low Power Modes

The VBAT-powered Control Interface has no low-power modes. It runs continuously while the VBAT well is powered.

26.8 General Description

The VBAT-Powered Control Interface (VCI) is used to drive the VCI_OUT pin. The output pin can be controlled either by VBAT-powered inputs, or by firmware when the VTR_CORE is active and the EC is powered and running. When the VCI_OUT pin is controlled by hardware, either because VTR_CORE is inactive or because the VCI block is configured for hardware control, the VCI_OUT pin can be asserted by a number of inputs:

- When one or more of the VCI_INx# pins are asserted. By default, the VCI_INx# pins are active low, but firmware can switch each input individually to an active-high input. See Section 26.8.1, "Input Polarity".
- When the Week Alarm from the Week Alarm Interface is asserted
- When the RTC Alarm from the Real Time Clock is asserted

Firmware can configure which of the hardware pin inputs contribute to the VCI_OUT output by setting the enable bits in the VCI Input Enable Register. Even if the input pins are not configured to affect VCI_OUT, firmware can monitor their current state through the status bits in the VCI Register. Firmware can also enable EC interrupts from the state of the input pins.

Each of the VCI_INx# pins can be configured for additional properties.

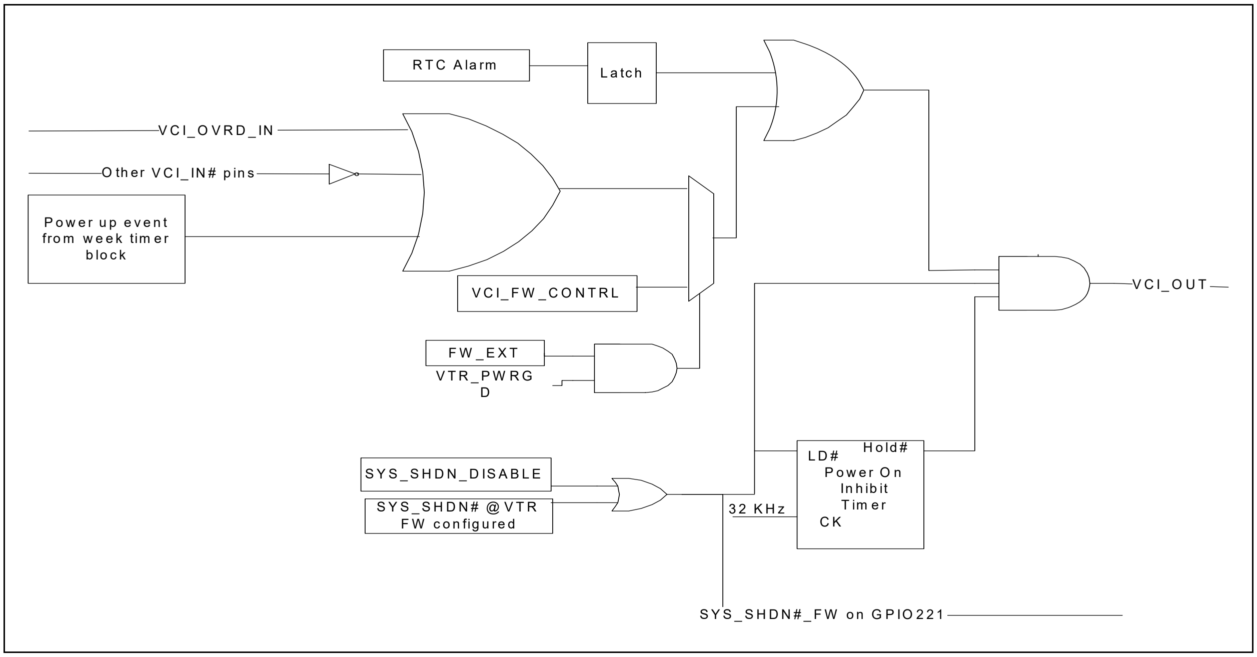
- By default, each of the VCI_INx# pins have an input glitch filter. All glitch filters can be disabled by the FILTERS_BYPASS bit in the VCI Register
- Assertions of each of the VCI_INx# pins can optionally be latched, so hardware can maintain the assertion of a VCI_INx# even after the physical pin is de-asserted, or so that firmware can determine which of the VCI_INx# inputs contributed to VCI_OUT assertion. See the Latch Enable Register and the Latch Resets Register.
- Rising edges and falling edges on the VCI_INx# pins are latched, so firmware can detect transitions on the VCI_INx# pins even if the transitions occurred while EC power was not available. See Section 26.8.2, "Edge Event Status".

If none of the additional properties are required, firmware can disable a VCI_INx# pin completely, by clearing both the corresponding bit in the VCI Input Enable Register and the corresponding bit in the VCI Buffer Enable Register. When both bits are ‘0’, the input is disabled and will not be a drain on the VBAT power rail.

When VTR_CORE power is present and the EC is operating, firmware can configure the VCI_OUT pin to operate as a general-purpose output pin. The VCI_OUT pin is firmware-controlled when the FW_EXT bit in the VCI Register is ‘1’. When firmware is controlling the output, the state of VCI_OUT is defined by the VCI_FW_CNTRL bit in the same register. When VTR_CORE is not present (the VTR_PWRGD input is low), the VCI_OUT pin is also determined by the hardware circuit.

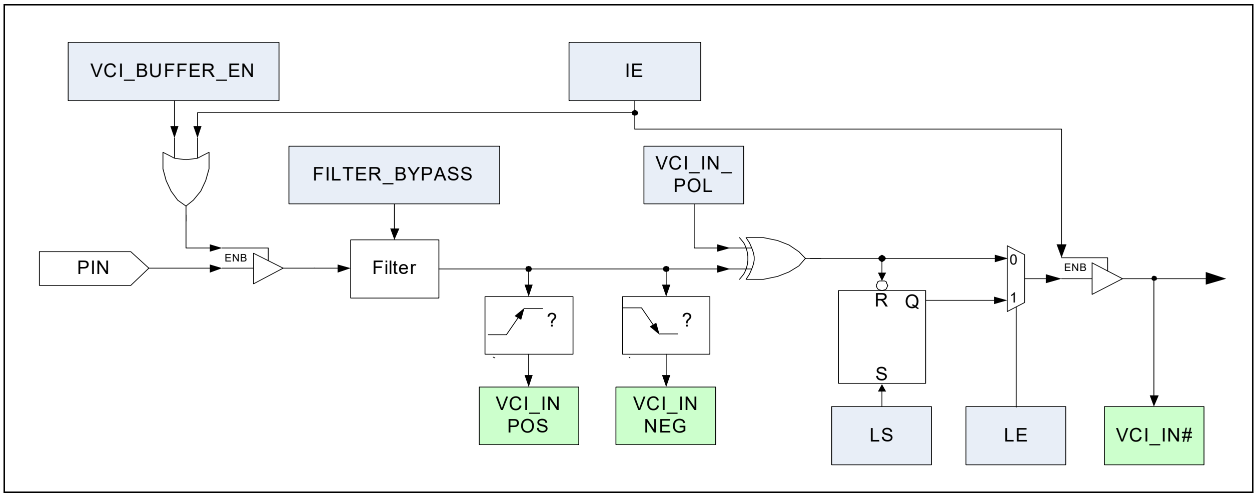
The following figures illustrate the VBAT-Power Control Interface logic:

FIGURE 26-2: VCI_OUT BLOCK DIAGRAM



The VCI_INx# Logic in the block diagram is illustrated in the following figure:

FIGURE 26-3: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



26.8.1 INPUT POLARITY

The VCI_INx# pins have an optional polarity inversion. The inversion takes place after any input filtering and before the VCI_INx# signals are latched in the VCI_INx# status bits in the VCI Register. Edge detection occurs before the polarity inversion. The inversion is controlled by battery-backed configuration bits in the VCI Polarity Register.

26.8.2 EDGE EVENT STATUS

Each VCI_INx# input pin is associated with two register bits used to record edge transitions on the pins. The edge detection takes place after any input filtering, before polarity control and occurs even if the VCI_INx# input is not enabled as part of the VCI_OUT logic (the corresponding control bit in the VCI Input Enable Register is ‘0’) or if the state of the

VCI_INx# input is not latched (the corresponding control bit in the Latch Enable Register is ‘0’). One bit is set whenever there is a high-to-low transition on the VCI_INx# pin (the VCI Negedge Detect Register) and the other bit is set whenever there is a low-to-high transition on the VCI_INx# pin (the VCI Posedge Detect Register).

In order to minimize power drain on the VBAT circuit, the edge detection logic operates only when the input buffer for a VCI_INx# pin is enabled. The input buffer is enabled either when the VCI_INx# pin is configured to determine the VCI_OUT pin, as controlled by the VCI_IN[1:0]# field of the VCI Register, or when the input buffer is explicitly enabled in the VCI Input Enable Register. When the pins are not enabled transitions on the pins are ignored.

26.8.3 VCI PIN MULTIPLEXING

Each of the VCI inputs, as well as VCI_OUT, are multiplexed with standard VTR_CORE-powered GPIOs. When VTR_CORE power is off, the mux control is disabled and the pin always reverts to the VCI function. The VCI_INx# function should be disabled in the VCI Input Enable Register VCI Buffer Enable Register and for any pin that is intended to be used as a GPIO rather than a VCI_INx#, so that VCI_OUT is not affected by the state of the pin.

26.8.4 POWER ON INHIBIT TIMER

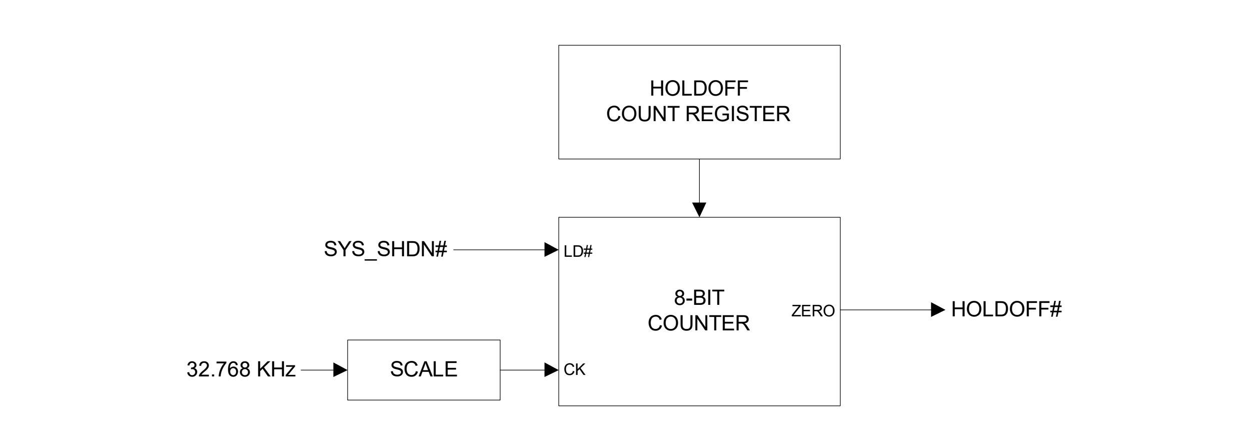
The Power On Inhibit Timer prevents the VBAT-Powered Control Interface VCI_OUT pin from being asserted for a programmable time period after the SYS_SHDN# pin asserted. This holdoff time can be used to give a system the opportunity to cool down after a thermal shutdown before allowing a user to attempt to turn the system on. While the Inhibit Timer is asserted, the VCI_OUT pin remains de-asserted and is unaffected by the VCI, Week Alarm and RTC interfaces.

The holdoff time is configured using the Holdoff Count Register. By setting the Holdoff Count Register to 0 the Inhibit Timer is disabled. When disabled, the HOLDOFF# signal is de-asserted and no counting takes place.

The HOLDOFF# output is asserted within one 32.768KHz clock cycle from the time SYS_SHDN# is asserted.

The following figure illustrates the operation of the Inhibit Timer:

FIGURE 26-4: POWER ON INHIBIT TIMER



The SCALE function reduces the 32.768KHz clock to 8Hz, so that the 8-bit counter counts intervals of 125ms. The following table shows some of examples of the effect of several settings of the Holdoff Count Register:

TABLE 26-3: HOLDOFF TIMING EXAMPLES

Holdoff Count Register	Holdoff Time (SEC)
0	Disabled (default)
1	0.125
5	0.625
10	1.25
15	1.875
100	12.5
150	18.75

TABLE 26-3: HOLDOFF TIMING EXAMPLES (CONTINUED)

Holdoff Count Register	Holdoff Time (SEC)
200	25
255	31.875

26.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the VBAT-Powered Control Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 26-4: REGISTER SUMMARY

EC Offset	Register Name
00h	VCI Register
04h	Latch Enable Register
08h	Latch Resets Register
0Ch	VCI Input Enable Register
10h	Holdoff Count Register
14h	VCI Polarity Register
18h	VCI Posedge Detect Register
1Ch	VCI Negedge Detect Register
20h	VCI Buffer Enable Register

26.9.1 VCI REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	Reserved	RES	-	-
18	MCHP reserved Should be set to zero		0	RESET _VBAT
17	RTC_ALRM If enabled by RTC_ALRM_LE, this bit is set to ‘1’ if the RTC Alarm signal is asserted. It is reset by writes to RTC_ALRM_LS.	R	0	RESET _VBAT
16	WEEK_ALRM If enabled by WEEK_ALRM_LE, this bit is set to ‘1’ if the Week Alarm signal is asserted. It is reset by writes to WEEK_ALRM_LS.	R	0	RESET _VBAT
15:13	Reserved	RES	-	-
Note 1:	The VCI_IN[3:0]# bits default to the state of their respective input pins. The VCI_OUT bit is determined by the VCI hardware circuit			

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
12	FILTERS_BYPASS The Filters Bypass bit is used to enable and disable the input filters on the VCI_INx# pins. See Section 47.17, "VBAT-Powered Control Interface Timing". 1=Filters disabled 0=Filters enabled (default)	R/W	0	RESET _VBAT
11	FW_EXT This bit controls selecting between the external VBAT-Powered Control Interface inputs, or the VCI_FW_CNTRL bit output to control the VCI_OUT pin. 1=VCI_OUT is determined by the VCI_FW_CNTRL field, when VTR_CORE is active **Note:** 0=VCI_OUT is determined by the external inputs.	R/W	0	RESET _SYS and RESET _VBAT
10	VCI_FW_CNTRL This bit can allow EC firmware to control the state of the VCI_OUT pin. For example, when VTR_PWRGD is asserted and the FW_EXT bit is '1', clearing the VCI_FW_CNTRL bit de-asserts the active high VCI_OUT pin. BIOS must set this bit to '1' prior to setting the FW_EXT bit to '1' on power up, in order to avoid glitches on the VCI_OUT pin.	R/W	0	RESET _SYS and RESET _VBAT
9	VCI_OUT This bit provides the current status of the VCI_OUT pin.	R	See Note 1	–
8	Reserved	RES	-	-
7	VCI_OUT/GPIO Select If this bit is 1 the signal is powered by VBAT and the output pin will be driven by VCI_OUT logic. If this bit is 0 the signal is powered by VTRx and the output pin will be driven according to GPIO Pin Control Registers. 1= VBAT Powered 0= VTR Powered	R/W	1	RESET _SYS and RESET _VBAT
6:4	Reserved	RES	-	-
3:0	VCI_IN[3:0]# These bits provide the latched state of the associated VCI_INx# pin, if latching is enabled or the current state of the pin if latching is not enabled. In both cases, the value is determined after the action of the VCI Polarity Register.	R	See Note 1	
Note 1:	The VCI_IN[3:0]# bits default to the state of their respective input pins. The VCI_OUT bit is determined by the VCI hardware circuit			

26.9.2 LATCH ENABLE REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:18	Reserved	RES	-	-
17	RTC_ALRM_LE Latch enable for the RTC Power-Up signal. 1=Enabled. Assertions of the RTC Alarm are held until the latch is reset by writing the correspondingLS[3:0] bit 0=Not Enabled. The RTC Alarm signal is not latched but passed directly to the VCI_OUT logic	R/W	0h	RESET _VBAT
16	WEEK_ALRM_LE Latch enable for the Week Alarm Power-Up signal. 1=Enabled. Assertions of the Week Alarm are held until the latch is reset by writing the correspondingLS[3:0] bit 0=Not Enabled. The Week Alarm signal is not latched but passed directly to the VCI_OUT logic	R/W	0h	RESET _VBAT
15:4	Reserved	RES	-	-
3:0	LE[3:0] Latching Enables. Latching occurs after the Polarity configuration, so a VCI_INx# pin is asserted when it is '0' if VCI_IN_POL[3:0] is '0', and asserted when it is '1 'if VCI_IN_POL[3:0] is '1'. For each LE[x] bit in the field: 1=Enabled. Assertions of the VCI_INx# pin are held until the latch is reset by writing the corresponding LS[3:0] bit 0=Not Enabled. The VCI_INx# signal is not latched but passed directly to the VCI_OUT logic	R/W	30h	RESET _VBAT

26.9.3 LATCH RESETS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:18	Reserved	RES	-	-
17	RTC_ALRM_LS RTC Alarm Latch Reset. When this bit is written with a '1', the RTC Alarm Event latch is reset The RTC Alarm input to the latch has priority over the Reset input Reads of this register are undefined.	W	-	–

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
16	WEEK_ALRM_LS Week Alarm Latch Reset. When this bit is written with a ‘1’, the Week Alarm Event latch is reset The Week Alarm input to the latch has priority over the Reset input Reads of this register are undefined.	W	-	–
15:4	Reserved	RES	-	-
3:0	LS[3:0] Latch Resets. When a Latch Resets bit (LS[x]) is written with a ‘1’, the corresponding VCI_IN*x*# latch is de-asserted (‘1’). The VCI_IN*x*# input to the latch has priority over the Latch Reset input, so firmware cannot reset the latch while the VCI_INx# pin is asserted. Firmware should sample the state of the pin in the VCI Register before attempting to reset the latch. As noted in the Latch Enable Register, the assertion level is determined by the VCI_IN_POL[3:0] bit. Reads of this register are undefined.	W	–	–

26.9.4 VCI INPUT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	IE[3:0] Input Enables for VCI_INx# signals. After changing the input enable for a VCI input, firmware should reset the input latch and clear any potential interrupt that may have been triggered by the input, as changing the enable may cause the internal status to change. For each IE[x] bit in the field: 1=Enabled. The corresponding VCI_INx# input is not gated and toggling the pin will affect the VCI_OUT pin 0=Not Enabled. The corresponding VCI_INx# input does not affect the VCI_OUT pin, even if the input is ‘0.’ Unless the corresponding bit in the VCI Buffer Enable Register is 1, latches are not asserted, even if the VCI_INx# pin is low, during a VBAT power transition	R/W	7h	RESET_VBAT

26.9.5 HOLDOFF COUNT REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	HOLDOFF_TIME These bits determine the period of time the VCI_OUT logic is inhibited from re-asserting VCI_OUT after a SYS_SHDN# event. FFh-01h=The Power On Inhibit Holdoff Time is set to a period between 125ms and 31.875 seconds. See Table 26-3 for examples 0=The Power On Inhibit function is disabled	RW	0	RESET_VBAT

26.9.6 VCI POLARITY REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	VCI_IN_POL[3:0] These bits determine the polarity of the VCI_INx input signals: For each VCI_IN_POL[x] bit in the field: 1=Active High. The value on the pins is inverted before use 0=Active Low (default)	RW	0	RESET_VBAT

26.9.7 VCI POSEDGE DETECT REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
3:0	VCI_IN_POS[3:0] These bits record a low to high transition on the VCI_INx# pins. A “1” indicates a transition occurred. For each VCI_IN_POS[x] bit in the field: 1=Positive Edge Detected 0=No edge detected	R/WC	0	RESET_VBAT

26.9.8 VCI NEGEDGE DETECT REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	VCI_IN_NEG[3:0] These bits record a high to low transition on the VCI_INx# pins. A "1" indicates a transition occurred. For each VCI_IN_NEG[x] bit in the field: 1=Negative Edge Detected 0=No edge detected	R/WC	0	RESET_VBAT

26.9.9 VCI BUFFER ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	VCI_BUFFER_EN[3:0] Input Buffer enable. After changing the buffer enable for a VCI input, firmware should reset the input latch and clear any potential interrupt that may have been triggered by the input, as changing the buffer may cause the internal status to change. This register has no effect when VTR_CORE is powered. When VTR_CORE is on, the input buffers are enabled only by the IE[3:0] bit. For each VCI_BUFFER_EN[x] bit in the field: 1=VCI_INx# input buffer enabled independent of the IE[3:0] bit. The edge detection latches for this input are always enabled 0=VCI_INx# input buffer enabled by the IE[3:0] bit. The edge detection latches are only enabled when the IE[3:0] bit is '1' (default)	RW	0	RESET_VBAT

27.0 VBAT-POWERED RAM

27.1 Overview

The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while the main power rail is operational, and will retain its values powered by battery power while the main rail is unpowered.

27.2 References

No references have been cited for this feature.

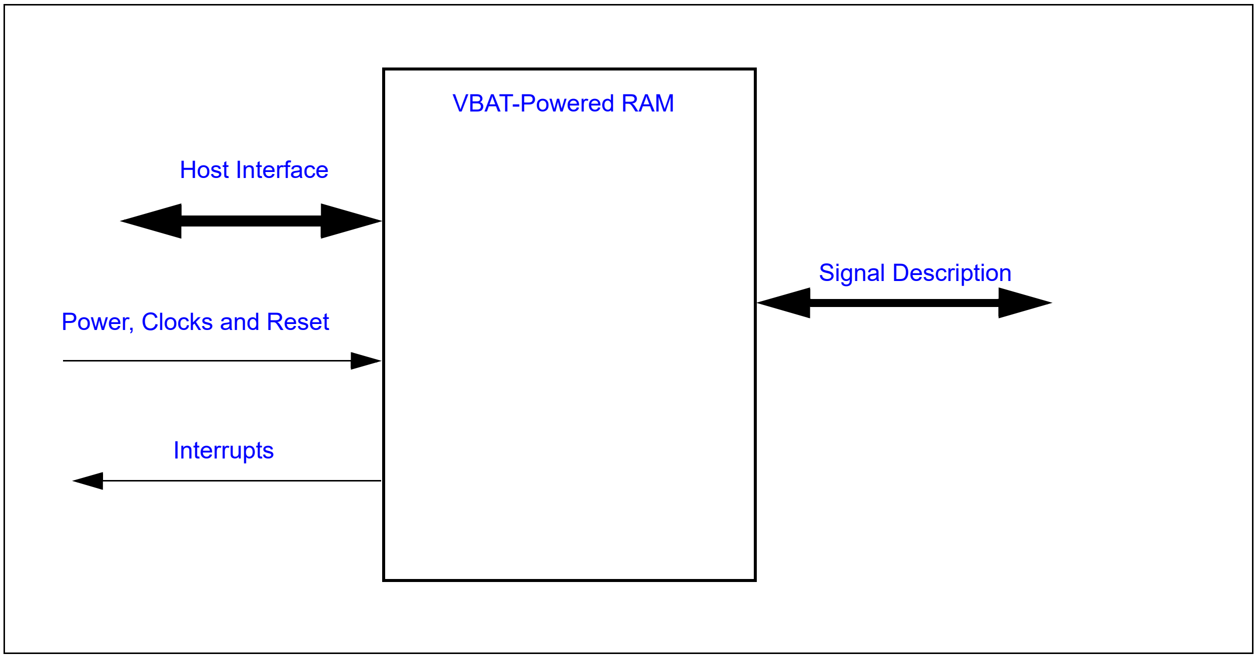
27.3 Terminology

There is no terminology defined for this section.

27.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 27-1: I/O DIAGRAM OF BLOCK



27.5 Signal Description

There are no external signals for this block.

27.6 Host Interface

The contents of the VBAT RAM are accessible only by the Embedded Controller (EC).

27.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

27.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The main power well used when the VBAT RAM is accessed by the EC.
VBAT	The power well used to retain memory state while the main power rail is unpowered.

27.7.2 CLOCK INPUTS

No special clocks are required for this block.

27.7.3 RESETS

Name	Description
RESET_VBAT	This signal resets all the registers and logic in this block to their default state.

27.8 Interrupts

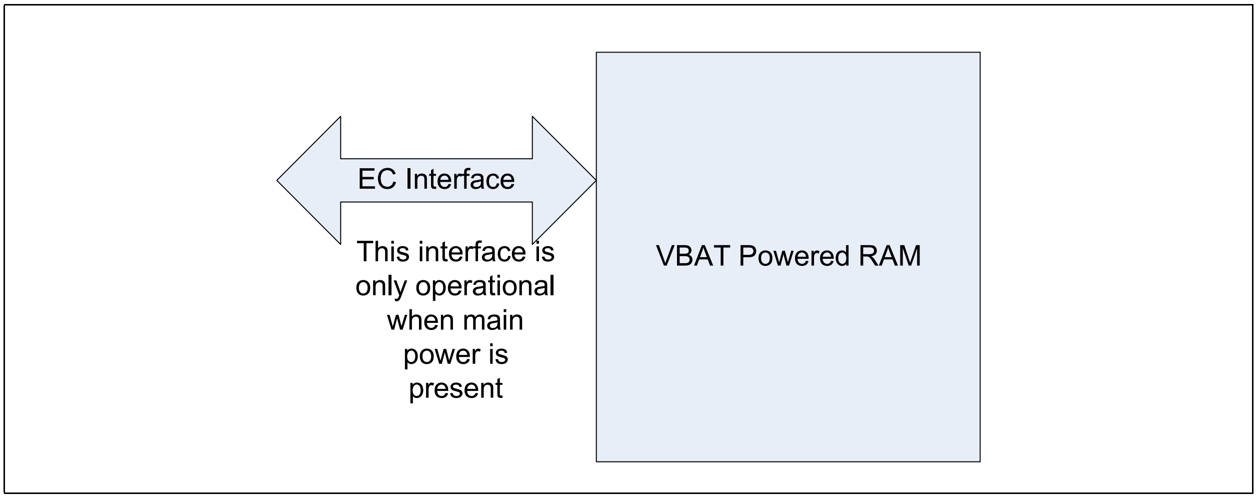
This block does not generate any interrupts.

27.9 Low Power Modes

The VBAT-Powered RAM automatically enters a low power mode whenever it is not being accessed by the EC. There is no chip-level Sleep Enable input.

27.10 Description

FIGURE 27-2: VBAT RAM BLOCK DIAGRAM


EC Interface
This interface is only operational when main power is present
VBAT Powered RAM

The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while VTR_CORE is powered, and will retain its values powered by VBAT while VTR_CORE is unpowered. The RAM is organized as a 16 words x 32-bit wide for a total of 64 bytes.

The contents of the VBAT RAM is indeterminate after a RESET_VBAT.

Note:	Any secret customer information stored on chip in VBAT memory must be encrypted for best security practices

28.0 VBAT REGISTER BANK

28.1 Introduction

This chapter defines a bank of registers powered by VBAT.

28.2 Interface

This block is designed to be accessed internally by the EC via the register interface.

28.3 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

28.3.1 POWER DOMAINS

Name	Description
VBAT	The VBAT Register Bank are all implemented on this single power domain.

28.3.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

28.3.3 RESETS

Name	Description
RESET_VBAT	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

28.4 Interrupts

This block does not generate any interrupt events.

28.5 Low Power Modes

The VBAT Register Bank is designed to always operate in the lowest power consumption state.

28.6 Description

The VBAT Register Bank block is a block implemented for aggregating miscellaneous battery-backed registers required the host and by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

28.7 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the VBAT Register Bank Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 28-1: REGISTER SUMMARY

Offset	Register Name
00h	Power-Fail and Reset Status Register
04h	TEST
08h	Clock Enable Register
0Ch	
10h	TEST
14h	

TABLE 28-1: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
1Ch	
20h	Monotonic Counter Register
24h	Counter HiWord Register
2Ch	TEST

28.7.1 POWER-FAIL AND RESET STATUS REGISTER

The Power-Fail and Reset Status Register collects and retains the VBAT RST and WDT event status when VTR_CORE is unpowered.

Address	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	VBAT_RST The VBAT_RST bit is set to ‘1’ by hardware when a RESET_VBAT is detected. This is the register default value. To clear VBAT RST EC firmware must write a ‘1’ to this bit; writing a ‘0’ to VBAT RST has no affect.	R/WC	1	RESET_ VBAT
6	SYSRESETREQ This bit is set to ‘1b’ if a RESET_SYS was triggered by an ARM SYSRESETREQ event. This bit is cleared to ‘0b’ when written with a ‘1b’; writes of a ‘0b’ have no effect.	R/WC	-	-
5	WDT This bit is set to ‘1b’ if a RESET_SYS was triggered by a Watchdog Timer event. This bit is cleared to ‘0b’ when written with a ‘1b’; writes of a ‘0b’ have no effect.	R/WC	0	RESET_ VBAT
4	RESETI This bit is set to ‘1b’ if a RESET_SYS was triggered by a low signal on the nRESET_IN input pin. This bit is cleared to ‘0b’ when written with a ‘1b’; writes of a ‘0b’ have no effect.	R/WC	0	RESET_ VBAT
3	TEST	R/WC	0	RESET_ VBAT
2	SOFT_SYS_RESET Status This bit is set to ‘1b’ if a was triggered by an assertion of the SOFT_SYS_RESET bit in the System Reset Register. This bit is cleared to ‘0b’ when written with a ‘1b’; writes of a ‘0b’ have no effect.	R/WC	0	RESET_ VBAT
1	Reserved	RES	-	-
0	Reserved	RES	-	-

28.7.2 CLOCK ENABLE REGISTER

Address	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2	32KHZ_SOURCE	R/W	0b	RESET_ VBAT
1	EXT_32K This bit selects the source for the 32KHz clock domain. 1=The 32KHZ_IN VTR-powered pin is used as a source for the 32KHz clock domain. If an activity detector does not detect a clock on the selected source, the 32KHz Clock internal clock source is automatically selected 0=The 32KHz Clock source is used as the source for the 32KHz clock domain	R/W	0b	RESET_ VBAT
0	Test	R	0b	-

28.7.3 MONOTONIC COUNTER REGISTER

Address	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MONOTTONIC_COUNTER Read-only register that increments by 1 every time it is read. It is reset to 0 on a VBAT Power On Reset.	R	0b	RESET_ VBAT

28.7.4 COUNTER HIWORD REGISTER

Address	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER_HIWORD Thirty-two bit read/write register. If software sets this register to an incrementing value, based on an external non-volatile store, this register may be combined with the Monotonic Counter Register to form a 64-bit monotonic counter.	R/W	0b	RESET_ VBAT

29.0 EC SUBSYSTEM REGISTERS

29.1 Introduction

This chapter defines a bank of registers associated with the EC Subsystem.

29.2 References

None

29.3 Interface

This block is designed to be accessed internally by the EC via the register interface.

29.4 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

29.4.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

29.4.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

29.4.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state, except WDT Event Count Register.
RESET_SYS_nWDT	This signal resets the WDT Event Count Register register. This reset is not asserted on a WDT Event.
RESET_VTR	This reset signal is asserted only on VTR_CORE power on.

29.5 Interrupts

This block does not generate any interrupt events.

29.6 Low Power Modes

The EC Subsystem Registers may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. When this block is commanded to sleep it will still allow read/write access to the registers.

29.7 Description

The EC Subsystem Registers block is a block implemented for aggregating miscellaneous registers required by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

29.8 EC-Only Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the EC Subsystem Registers Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 29-1: REGISTER SUMMARY

Offset	Register Name
00h	Reserved
04h	AHB Error Address Register
08h	TEST
0Ch	TEST
10h	TEST
14h	AHB Error Control Register
18h	Interrupt Control Register
1Ch	ETM TRACE Enable Register
20h	Debug Enable Register
28h	WDT Event Count Register
2Ch	TEST
30h	TEST
34h	TEST
38h	Reserved
3Ch	TEST
40h	PECI Disable Register
44h	TEST
48h	TEST
54h	TEST
5Ch	TEST
60h	TEST
64h	GPIO Bank Power Register
68h	TEST
6Ch	TEST
90h	Virtual Wire Source Configuration Register
100h	TEST

29.8.1 AHB ERROR ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	AHB_ERR_ADDR In priority order: 1. AHB address is registered when an AHB error occurs on the processors AHB master port and the register value was already 0. This way only the first address to generate an exception is captured. 2. The processor can clear this register by writing any 32-bit value to this register.	R/WZC	0h	RESET_ SYS

29.8.2 AHB ERROR CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	TEST	R/W	0h	RESET_ SYS
0	AHB_ERROR_DISABLE 1=EC memory exceptions are disabled 0=EC memory exceptions are enabled	R/W	0h	RESET_ SYS

29.8.3 INTERRUPT CONTROL REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	NVIC_EN This bit enables Alternate NVIC IRQ's Vectors. The Alternate NVIC Vectors provides each interrupt event with a dedicated (direct) NVIC vector. 1=Alternate NVIC vectors enabled 0=Alternate NVIC vectors disabled	R/W	1b	RESET_ SYS

29.8.4 ETM TRACE ENABLE REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	TRACE_EN This bit enables the ARM TRACE debug port (ETM/ITM). The Trace Debug pins are forced to the TRACE functions. 1=ARM TRACE port enabled 0=ARM TRACE port disabled	R/W	0b	RESET_ SYS

29.8.5 DEBUG ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	RES	-	-
4	BOUNDARY SCAN PORT ENABLE 1= Enable Boundary scan port enable The Boundary Scan Tap controller is accessible via JTAG Port 0= Disable Boundary scan port enable The Boundary scan Tap controller is not accessible via JTAG Port. If JTAG_STRAP pin is sampled low, this bit cannot enable Boundary scan mode. Please refer to TAP Controller Select Strap Option for usage of this bit.	R/W	0h	RESET_SYS
3	DEBUG_PU_EN If this bit is set to '1b' internal pull-up resistors are automatically enabled on the appropriate debugging port wires whenever the debug port is enabled (the DEBUG_EN bit in this register is '1b' and the JTAG_RST# pin is high). The setting of DEBUG_PIN_CFG determines which pins have pull-ups enabled when the debug port is enabled.	R/W	0h	RESET_SYS
2:1	DEBUG_PIN_CFG This field determines which pins are affected by the JTAG_RST# debug enable pin. 3=Reserved 2=The pins associated with the JTAG TCK and TMS switch to the debug interface when JTAG_RST# is de-asserted high. The pins associated with TDI and TDO remain controlled by the associated GPIO. This setting should be used when the ARM Serial Wire Debug (SWD) is required for debugging and the Serial Wire Viewer is not required 1=The pins associated with the JTAG TCK, TMS and TDO switch to the debug interface when JTAG_RST# is de-asserted high. The pin associated with TDI remains controlled by the associated GPIO. This setting should be used when the ARM Serial Wire Debug (SWD) and Serial Wire Viewer (SWV) are both required for debugging 0=All four pins associated with JTAG (TCK, TMS, TDI and TDO) switch to the debug interface when JTAG_RST# is de-asserted high. This setting should be used when the JTAG TAP controller is required for debugging	R/W	0h	RESET_SYS

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	DEBUG_EN This bit enables the JTAG/SWD debug port. 1=JTAG/SWD port enabled. A high on JTAG_RST# enables JTAG or SWD, as determined by SWD_EN 0=JTAG/SWD port disabled. JTAG/SWD cannot be enabled (the JTAG_RST# pin is ignored and the JTAG signals remain in their non-JTAG state) If JTAG_RST# pin is sampled low, this bit cannot enable JTAG mode.	R/W	0b	RESET_SYS

Note: Boot ROM updates this register value on exit. Refer to the Boot ROM document for details.

29.8.6 WDT EVENT COUNT REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	WDT_EVENT_COUNT This field is cleared to 0 on a reset triggered by the main power on reset, but not on a reset triggered by the Watchdog Timer. This field needs to be written by application to indicate the number of times a WDT fired before loading a good EC code image. Note 1	R/W	0b	RESET_SYS_n-WDT

Note 1: The recommended procedure is to first clear the WDT Status Register followed by incrementing the WDT_EVENT_COUNT.

29.8.7 PECI DISABLE REGISTER

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	PECI_DISABLE This bit reduces leakage current through the CPU voltage reference pin if PECI or SB-TSI are not used. 1=The VREF_VTT function is disabled, independent of the mux setting of the GPIO that shares the pin. The GPIO that shares the pin is not disabled 0=The VREF_VTT pin is enabled	R/W	0b	RESET_SYS

29.8.8 GPIO BANK POWER REGISTER

Offset	64h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	–

7	GPIO Bank Power Lock
	0 = VTR_LEVEL bits[2:0] and GPIO Bank Power Lock bit are R/W
	1 = VTR_LEVEL bits[2:0] and GPIO Bank Power Lock bit are Read Only
	This bit cannot be cleared once it is set to '1'. Writing zero has no effect.
		Bit[7]=0 R/W
		Bit[7]=1 RO
			0h	RESET_SYS

6:3	Reserved	RES	-	–
1	VTR_LEVEL2 Voltage level on VTR2 power rail. Software will set this bit based on the VTR2_STAP pin. 1=VTR2 is powered by 1.8V 0=VTR2 is powered by 3.3V	see Bit[7]	0h	RESET_SYS
0	TEST This is a TEST bit and should be always programmed to 0b for proper functioning of the device.	RW	0h	RESET_SYS

Note: The Boot ROM reads the VTR_LEVEL2 values from the SPI Flash Header and writes the VTR_LEVEL2,bits. If the SPI Flash load fails, the Boot ROM clears all VTR_LEVEL2, bits.

29.8.9 VIRTUAL WIRE SOURCE CONFIGURATION REGISTER

Offset	90h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
1:0	VWIRE_SOURCE VWIRE_SOURCE [2] should always be programmed to 1b. VWIRE_SOURCE [1] 0 = The hardware source MBX_Host_SMI affects the state of the SMI# (SRC1) bit of the SMVW02 register. 1 = The hardware source MBX_Host_SMI does not affect the SMI# (SRC1) bit of the SMVW02 register. **Note:** Firmware can always write to the SRC1 bit of the SMVW02 register. VWIRE_SOURCE [0] 0=The hardware source EC_SCI# affects the state of the SCI# (SRC0) bit of the SMVW02 register. 1= The hardware source EC_SCI# does not affect the SCI# (SRC0) bit of the SMVW02 register. **Note:** Firmware can always write to the SRC0 bit of the SMVW02 register.	RW	7h	RESET_SYS

30.0 SECURITY FEATURES

30.1 Overview

This device includes a set of components that can support a high level of system security. Hardware support is provided for:

- Authentication, using public key algorithms
- Integrity, using Secure Hash Algorithms (SHA)
- Privacy, using symmetric encryption (Advanced Encryption Standard, AES)
- Entropy, using a true Random Number Generator

30.2 References

- American National Standards Institute, “Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography”, X9.63-2011, December 2011
- American National Standards Institute, “Public Key Cryptography for the Financial Servic3es Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)”, X9.62-2005, November 2005
- International Standards Organization, “Information Technology - Security techniques - Cryptographic techniques based on elliptic curves -- Part 2: Digital Signatures”, ISO/IEC 15946-2, December 2002
- National Institute of Standards and Technology, “Secure Hash Standard (SHS)”, FIPS Pub 180-4, March 2012
- National Institute of Standards and Technology, “Digital Signature Standard (DSS)”, FIPS Pub 186-3, June 2009
- National Institute of Standards and Technology, “Advanced Encryption Standard (AES)”, FIPS Pub 197, November 2001
- National Institute of Standards and Technology, “Recommendation for Block Cipher Modes of Operation”, FIPS SP 800-38A, 2001
- RSA Laboratories, “PKCS#1 v2.2: RSA Cryptography Standard”, October 2012

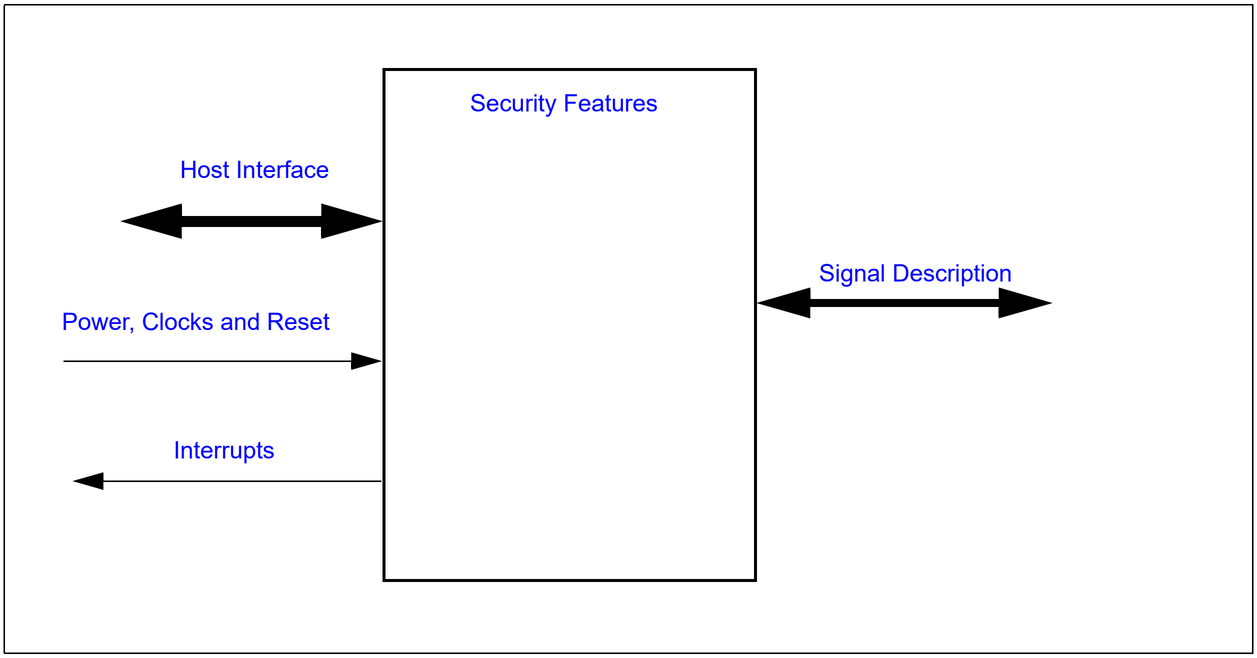
30.3 Terminology

There is no terminology defined for this section.

30.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 30-1: I/O DIAGRAM OF BLOCK



30.5 Signal Description

There are no external signals for this block.

30.6 Host Interface

Registers for the cryptographic hardware are accessible by the EC.

30.7 Power, Clocks and Reset

30.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The main power well used when the VBAT RAM is accessed by the EC.

30.7.2 CLOCK INPUTS

No special clocks are required for this block.

30.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

30.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
Public Key Engine	
PKE_ERROR	Public Key Engine core error detected
PKE END	Public Key Engine completed processing
Symmetric Encryption	
AES	Symmetric Encryption block completed processing
Cryptographic Hashing	
HASH	HASH
Random Number Generator	
RNG	Random Number Generator filled its FIFO

30.9 Low Power Modes

The Security Features may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

30.10 Description

The security hardware incorporates the following functions:

30.10.1 SYMMETRIC ENCRYPTION/DECRYPTION

Standard AES encryption and decryption, with key sizes of 128 bits, 192 bits and 256 bits, are supported with a hardware accelerator. AES modes that can be configured include Electronic Code Block (ECB), Cipher Block Chaining (CBC), Counter Mode (CTR), Output Feedback (OFB) and Cipher Feedback (CFB).

30.10.2 CRYPTOGRAPHIC HASHING

Standard SHA hash algorithms, including SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512, are supported by hardware.

30.10.3 PUBLIC KEY CRYPTOGRAPHIC ENGINE

A large variety of public key algorithms are supported directly in hardware. These include:

- RSA encryption and decryption, with key sizes of 1024 bits, 2048 bits, 3072 bits and 4096 bits
- Elliptic Curve point multiply, with all standard NIST curves, using either binary fields or prime fields
- Elliptic Curve point multiply with Curve25519
- The Elliptic Curve Digital Signature Algorithm (ECDSA), using all supported NIST curves
- The Elliptic Curve Korean Certificate-based Digital Signature Algorithm (EC-KCDSA), using all supported NIST curves
- The Edwards-curve Digital Signature Algorithm (EdDSA), using Curve25519
- Miller-Rabin primality testing

The Public Key Engine includes a 24KB cryptographic SRAM, which can be accessed by the EC when the engine is not in operation. With its private SRAM memory, the Public Key Engine can process public key operations independently of the EC.

30.10.4 TRUE RANDOM NUMBER GENERATOR

A true Random Number Generator, which includes a 1K bit FIFO for pre-calculation of random bits.

30.10.5 MONOTONIC COUNTER

The Monotonic Counter is defined in Section 28.7.3, "Monotonic Counter Register". The counter automatically increments every time it is accessed, as long as VBAT power is maintained. If it is necessary to maintain a monotonic counter across VBAT power cycles, the Counter HiWord Register can be combined with the Monotonic Counter Register to form a 64-bit monotonic counter. Firmware would be responsible for updating the Counter HiWord on a VBAT POR. The HiWord could be maintained in a non-volatile source, such as the EEPROM or an external SPI Flash.

30.10.6 CRYPTOGRAPHIC API

The Boot ROM includes an API for direct software access to cryptographic functions. API functions for Hashing and AES include a DMA interface, so the operations can function on large blocks of SRAM with a single call.

30.11 Registers

TABLE 30-1: CRYPTOGRAPHIC SRAM

Block Instance	Start Address	End Address	Size
Cryptographic SRAM	4010_0000h	4010_5FFF	24KB

31.0 OTP BLOCK

31.1 Introduction

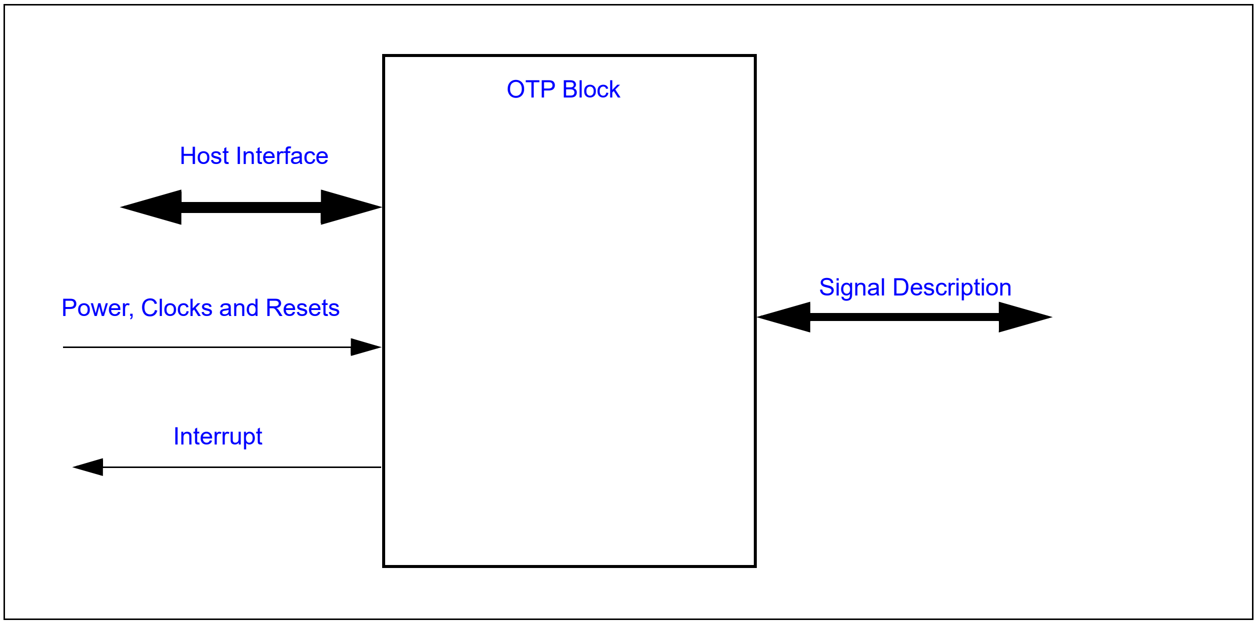
The OTP Block provides a means of programming and accessing a block of One Time Programmable memory.

31.2 Terminology

None.

31.3 Interface

FIGURE 31-1: OTP BLOCK INTERFACE DIAGRAM



31.4 Signal Description

There are no external signals from this block

31.5 Host Interface

The registers defined for the OTP Block are accessible by the EC.

31.6 Interrupt Interface

TABLE 31-1: INTERRUPT SIGNALS

Source	Description
OTP_READY	The OTP_READY interrupt will be generated whenever an OTP command is completed.

31.7 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

31.7.1 POWER DOMAINS

TABLE 31-2: POWER SOURCES

Name	Description
VTR_CORE	This power well sources all of the registers and logic in this block, except where noted.
VTR	This is the IO voltage for the block.

31.7.2 CLOCKS

This section describes all the clocks in the block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

TABLE 31-3: CLOCKS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

31.7.3 RESETS

TABLE 31-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.

31.8 Low Power Modes

The OTP always comes up in low power mode and stays in that state unless the firmware needs to use it

31.9 Description

The OTP Block has a capacity of 8 K bits arranged as 1K x 8 bits.

Note: Any secret customer information stored on chip in OTP memory must be encrypted for best security practices

31.10 OTP Memory Map

Please refer to Boot ROM document for this information.

TABLE 31-5: REGISTER SUMMARY

Offset	Register Name
44h	OTP Write Lock Register
48h	OTP Read Lock Register

31.10.1 OTP WRITE LOCK REGISTER

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	OTP_WRLOCK When any of these bits are set, the corresponding 32 byte range in the OTP is not writable.	R/W1S	0h	RESET_SYS

31.10.2 OTP READ LOCK REGISTER

Offset	48h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	OTP_RDLOCK When any of these bits are set, the corresponding 32 byte range in the OTP is not readable.	R/W1S	0h	RESET_SYS

Note 1: OTP Memory can be locked by writing to OTP bytes 1012 - 1019. Boot ROM will then lock the region on every Boot preventing the code that is loaded from accessing this memory location.

2: Application FW can write to the above lock registers and lock the memory region preventing other code loaded from accessing the locked region. This is useful in multistage boot loaders

32.0 TEST MECHANISMS

32.1 ARM Test Functions

Test mechanisms for the ARM are described in Section 5.0, "ARM M4 Based Embedded Controller".

32.2 JTAG Boundary Scan

Note: Boundary Scan operates in 4-wire JTAG mode only. This is not supported by 2-wire SWD.

JTAG Boundary Scan includes registers and functionality as defined in IEEE 1149.1 and the CEC1712 BSDL file. Functionality implemented beyond the standard definition is summarized in Table 32-2. The CEC1712 Boundary Scan JTAG ID is shown in Table 1-1.

Note: Must wait a minimum of 35ms after a POR to accurately read the Boundary Scan JTAG ID. Reading the JTAG ID too soon may return a Boundary Scan JTAG ID of 00000000h. This is not a valid ID value.

32.2.1 TAP CONTROLLER SELECT STRAP OPTION

The TAP Controller Select Strap Option determines the JTAG slave that is selected when JTAG_RST# is not asserted. The state of the TAP Controller Select Strap Option pin, defined in the Pin Configuration chapter (JTAG_STRAP), is sampled by hardware at POR according to the Slave Select Timing as defined in Section 34.14, "JTAG Interface Timing" The same JTAG port is used for accessing boundary scan and ARM TAP Controller based on the JTAG_STRAP pin.

TABLE 32-1: illustrates the selection of the Tap controller over the JTAG port

By default the ARM Tap controller defaults to 2-Pin mode and may be configured to 4- pin

TABLE 32-1: TAP CONTROLLER SELECTION

JTAG_STRAP Pin Value *at Power On*	**JTAG_RST#**	BOUNDARY SCAN PORT ENABLE **Bit**	DEBUG_EN **(JTAG Debug Disable) Bit**	**Description**
0	0	X	x	Boundary scan/ ARM Tap controller cannot be accessed through JTAG port
0	1	X	0	Hardware does not enable Boundary scan functionality. BOUNDARY SCAN PORT ENABLE bit has no effect. ARM TAP controller cannot be accessed
0	1	X	1	Hardware does not enable Boundary scan functionality. BOUNDARY SCAN PORT ENABLE bit has no effect. ARM TAP controller can be accessed
1	0	x	x	Hardware does not enable Boundary Scan functionality Boundary Scan TAP controller or ARM TAP controller is not accessible via JTAG port.

JTAG_STRAP Pin Value *at Power On*	JTAG_RST#	BOUNDARY SCAN PORT ENABLE Bit	DEBUG_EN (JTAG Debug Disable) Bit	Description
1	1	0	x	Hardware enables Boundary Scan functionality, but Boundary Scan Port is disabled. Boundary Scan TAP controller or ARM TAP controller is not accessible via JTAG port.
1	1	1	x	Hardware enables Boundary Scan functionality and Boundary Scan Port is enabled. Boundary Scan TAP controller is accessible via JTAG port ARM TAP controller is not accessible via JTAG port

TABLE 32-2: EXTENDED BOUNDARY SCAN FUNCTIONALITY

Bits	Function	Description
12, 14	TAP Controller Select Strap Option Override	When the Strap Option Override is ‘1,’ the strap option is overridden to select the debug TAP Controller until the next time the strap is sampled. To set Strap Override Function, write 0X1FFFFD to the TAP controller instruction register, then write 0x5000 to the TAP controller data register. Note that the instruction register is 18 bits long; the data register is 16 bits long.

33.0 ELECTRICAL SPECIFICATIONS

33.1 Maximum Ratings*

*Stresses exceeding those listed could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

> **Note:** When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

33.1.1 ABSOLUTE MAXIMUM THERMAL RATINGS

Parameter	Maximum Limits
Operating Temperature Range	-40^{o}C to +85^{o}C Industrial
Storage Temperature Range	-55^{o} to +150^{o}C
Lead Temperature Range	Refer to JEDEC Spec J-STD-020B

33.1.2 ABSOLUTE MAXIMUM SUPPLY VOLTAGE RATINGS

Symbol	Parameter	Maximum Limits
VBAT	3.0V Battery Backup Power Supply with respect to ground	-0.3V to +3.63V
VTR_REG	Main Regulator Power Supply with respect to ground	-0.3V to +3.63V
VTR_ANALOG	3.3V Analog Power Supply with respect to ground	-0.3V to +3.63V
VTR1	3.3V Power Supply with respect to ground	-0.3V to +3.63V
VTR2	3.3V or 1.8V Power Supply with respect to ground	-0.3V to +3.63V

33.1.3 ABSOLUTE MAXIMUM I/O VOLTAGE RATINGS

Parameter	Maximum Limits
Voltage on any Digital Pin with respect to ground	Determined by Power Supply of I/O Buffer and Pad Type

33.2 Operational Specifications

33.2.1 POWER SUPPLY OPERATIONAL CHARACTERISTICS

TABLE 33-1: POWER SUPPLY OPERATING CONDITIONS

Symbol	Parameter	MIN	TYP	MAX	Units
VBAT	Battery Backup Power Supply	2.0	3.0	3.465	V
VTR_REG	Main Regulator Power Supply	1.71	3.0	3.465	V
VTR_ANALOG	Analog Power Supply	3.135	3.3	3.465	V
VTRx	3.3V Power Supply	3.135	3.3	3.465	V
	1.8V Power Supply	1.71	1.80	1.89	V

Note: The specification for the VTRx supplies are +/- 5%.

33.2.2 AC ELECTRICAL SPECIFICATIONS

The AC Electrical Specifications for the clock input time are defined in Section 34.4, "Clocking AC Timing Characteristics". The clock rise and fall times use the standard input thresholds of 0.8V and 2.0V unless otherwise specified and the capacitive values listed in this section.

33.2.3 CAPACITIVE LOADING SPECIFICATIONS

The following table defines the maximum capacitive load validated for the buffer characteristics listed in Table 33-3, "DC Electrical Characteristics" and the AC characteristics defined in Section 34.4, "Clocking AC Timing Characteristics".

CAPACITANCE T_A = 25°C; fc = 1MHz; V_{cc} = 3.3 VDC

Note: All output pins, except pin under test, tied to AC ground.

TABLE 33-2: MAXIMUM CAPACITIVE LOADING

Parameter	Symbol	Limits			Unit	Notes
		MIN	TYP	MAX		
Input Capacitance of PECI_IO	C_{IN}			10	pF	
Output Load Capacitance supported by PECI_IO	C_{OUT}			10	pF	
Input Capacitance (all other input pins)	C_{IN}			10	pF	Note 1
Output Capacitance (all other output pins)	C_{OUT}			20	pF	Note 2

Note 1: All input buffers can be characterized by this capacitance unless otherwise specified.

2: All output buffers can be characterized by this capacitance unless otherwise specified.

33.2.4 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 33-3: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO Type Buffer						
All PIO Buffers						Internal PU selected via the GPIO Pin Control Register.
Pull-up Resistor	R_{PU}					
@3.3V		34	52	95	KΩ	
@1.8V		35	60	105		
All PIO Buffers						Internal PD selected via the GPIO Pin Control Register.
Pull-down Resistor	R_{PD}				KΩ	
@3.3V		38	63	127		
@1.8V		36	63	118		
PIO						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2
DRIVE_STRENGTH = 00b	_				_	Same characteristics as an IO-2 mA.
DRIVE_STRENGTH = 01b	_				_	Same characteristics as an IO-4 mA.
DRIVE_STRENGTH = 10b	_				_	Same characteristics as an IO-8 mA.
DRIVE_STRENGTH = 11b	_				_	Same characteristics as an IO-12 mA.
I Type Input Buffer						TTL Compatible Schmitt Trigger Input
Low Input Level	V_{ILI}			0.3x VTR	V	
High Input Level	V_{IHI}		0.7x VTR		V	
Schmitt Trigger Hysteresis	V_{HYS}		400		mV	
O-2 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 2 mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -2 mA (min)
IO-2 mA Type Buffer	_				_	Same characteristics as an I and an O-2mA.
OD-2 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 2 mA (min)

TABLE 33-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
IOD-2 mA Type Buffer	–				–	Same characteristics as an I and an OD-2mA.
O-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -4 mA (min)
IO-4 mA Type Buffer	–				–	Same characteristics as an I and an O-4mA.
OD-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA (min)
IOD-4 mA Type Buffer	–				–	Same characteristics as an I and an OD-4mA.
O-8 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -8 mA (min)
IO-8 mA Type Buffer	–				–	Same characteristics as an I and an O-8mA.
OD-8 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA (min)
IOD-8 mA Type Buffer	–				–	Same characteristics as an I and an OD-8mA.
O-12 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -12mA (min)
IO-12 mA Type Buffer	–				–	Same characteristics as an I and an O-12mA.
OD-12 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA (min)
IOD-12 mA Type Buffer	–				–	Same characteristics as an I and an OD-12mA.

TABLE 33-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
I_AN Type Buffer						
I_AN Type Buffer (Analog Input Buffer)	I_AN	Voltage range on pins: -0.3V to +3.63V				These buffers are not 5V tolerant buffers and they are not back-drive protected
ADC Reference Pins						
VREF_ADC						
Voltage (Option A)	V		VTR		V	Connect to same power supply as VTR
Voltage (Option B)	V	2.97	3.0	3.03	V	
Input Impedance	R_{REF}		75		KΩ	
Input Low Current	ILEAK	-0.05		+0.05	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.
Note: Tolerance for the pins are not 5VT Unless the pin chapter explicitly indicates specific pin has “Over-voltage protection” feature.						

33.2.4.1 Pin Leakage

Leakage characteristics for all digital I/O pins is shown in the following Pin Leakage table, unless otherwise specified. Two exceptions are pins with Over-voltage protection and Backdrive protection. Leakage characteristics for Over-Voltage protected pins and Backdrive protected pins are shown in the two sub-sections following the Pin Leakage table.

TABLE 33-4: PIN LEAKAGE (VTR=3.3V + 5%; VTR = 1.8V +5%)

(TA = -40°C to +85°C)						
Parameter	**Symbol**	**MIN**	**TYP**	**MAX**	**Units**	**Comments**
Leakage Current	I_{IL}	-2		+2	µA	VIN=0V to VTR

OVER-VOLTAGE PROTECTION TOLERANCE

Note: 5V tolerant pins have both backdrive protection and over-voltage protection.

All the I/O buffers that do not have “Over-voltage Protection” are can only tolerate up to +/-10% I/O operation (or +1.98V when powered by 1.8V, or 3.63V when powered by 3.3V).

Functional pins that have “Over-voltage Protection” can tolerate up to 3.63V when powered by 1.8V, or 5.5V when powered by 3.3V. These pins are also backdrive protected. Backdrive Protection characteristics are shown in the following table:

TABLE 33-5: 5V TOLERANT LEAKAGE CURRENTS (VTR=3.3-5%)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
(T_A = -40°C to +85°C)						
Three-State Input Leakage Current for 5V Tolerant Pins	I_{IL}	-3		3	µA	Vin = 0 to 3.5V
		-130		130		Vin = 3.5V to 4.1V
		-3		3		Vin = 4.1 to 5V

Note: These measurements are done without an external pull-up.

TABLE 33-6: 3.6V TOLERANT LEAKAGE CURRENTS (VTR = 1.8V-5%)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
(T_A = -40°C to +85°C)						
Three-State Input Leakage Current for Under-Voltage Tolerant Pins	I_{IL}	-2		2	µA	Vin = 0 to 2.1V
		-25		25		Vin = 2.1V to 2.5V
		-2		2		Vin = 3.2V to 3.6V

Note: This measurements are done without an external pull-up.

BACKDRIVE PROTECTION

TABLE 33-7: BACKDRIVE PROTECTION LEAKAGE CURRENTS (VTR=0V)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
(T_A = -40°C to +85°C)						
Input Leakage	I_{IL}	-1		2	µA	VIN@3.47V
Input Leakage	I_{IL}	-2		9	µA	VIN@5.0V

33.2.5 ADC ELECTRICAL CHARACTERISTICS

TABLE 33-8: ADC CHARACTERISTICS

Symbol	Parameter	MIN	TYP	MAX	Units	Comments
VTR_ ANALOG	Analog Supply Voltage (powered by VTR)	3.135	3.3	3.465	V	
V_{RNG}	Input Voltage Range	0		VREF _ADC	V	Range of VREF_ADC input to ADC ground
RES	Resolution	–	–	10/12	Bits	Guaranteed Monotonic

TABLE 33-8: ADC CHARACTERISTICS (CONTINUED)

Symbol	Parameter	MIN	TYP	MAX	Units	Comments
ACC	Absolute Accuracy	–	2	4	LSB	
DNL	Differential Non Linearity, DNL	-1	–	+1	LSB	Guaranteed Monotonic
INL	Integral Non Linearity, INL	-3.0	–	+3	LSB	Guaranteed Monotonic
E_{GAIN}	Gain Error, E_{GAIN}	-2	–	2	LSB	
E_{OFFSET}	Offset Error, E_{OFFSET}	-2	–	2	LSB	
CONV	Conversion Time		1.125		µS/channel	
II	Input Impedance	4	4.5	5.3	MΩ	

33.2.6 THERMAL CHARACTERISTICS

TABLE 33-9: THERMAL OPERATING CONDITIONS

Rating	Symbol	MIN	TYP	MAX	Unit
Consumer Temperature Devices					
Operating Junction Temperature Range	T_J	0	—	125	°C
Operating Ambient Temperature Range - Industrial	T_A	-40	—	+85	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} – S\ I_{OH})$ I/O Pin Power Dissipation: $I/O = S\ ((\{V_{DD} – V_{OH}\} \times I_{OH}) + S\ (V_{OL} \times I_{OL}))$	P_D	69.3 ($P_{INT} + P_{I/O}$)			mW
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J$[a]$– T_A)/\theta_{JA}$			W

[a] Tj Max value is at ambient of 70°C.

Note: Junction to ambient thermal resistance, Theta-JA (θ_{JA}), and Junction to case thermal resistance, Theta-JC (θ_{JC}), numbers are achieved by package simulations.

33.3 Power Consumption

TABLE 33-11: VTR SUPPLY CURRENT, I_VTR

VCC	VTR	48 MHz Clock	VTR1			VTR2			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On	On	48MHz	0.15	0.16	0.17	0.01	0.02	0.02	mA	Full On
On	On	PLL Gated	0.15	0.16	0.17	0.01	0.02	0.02	mA	Light Sleep
On	On	PLL Off	0.04	0.05	0.06	0.01	0.02	0.02	mA	Heavy Sleep
Off	On	48MHz	0.15	0.16	0.17	0.01	0.02	0.02	mA	Full On
Off	On	PLL Gated	0.15	0.16	0.17	0.01	0.02	0.02	mA	Light Sleep
Off	On	PLL Off	0.04	0.05	0.06	0.01	0.02	0.02	mA	Heavy Sleep

TABLE 33-12: VTR SUPPLY CURRENT, I_VTR

VCC	VTR	48 MHz Clock	VTR_REG			VTR_PLL			VTR_ANALOG			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On	On	48MHz	7.5	11.5	13.5	0.10	0.11	0.11	1.2	1.5	1.5	mA	Full On
On	On	PLL Gated	2.0	5.5	7.0	0.1	0.11	0.11	0.15	0.15	0.17	mA	Light Sleep
On	On	PLL Off	0.2	3.5	5.25	0.01	0.01	0.01	0.15	0.15	0.17	mA	Heavy Sleep
Off	On	48MHz	7.5	11.5	13.5	0.10	0.11	0.11	1.2	1.5	1.5	mA	Full On
Off	On	PLL Gated	2.0	5.5	7.0	0.10	0.11	0.11	0.15	0.15	0.17	mA	Light Sleep
Off	On	PLL Off	0.2	3.5	5.25	0.01	0.01	0.01	0.15	0.15	0.17	mA	Heavy Sleep

Note 1: Full On is defined as follows: The processor is not sleeping, the PLL is powered and the following blocks are Active: ADC, EC Subsystem, Hibernation Timer, Interrupt Controller, PWM, TFDP, Basic Timers, JTAG, RTC.

2: The sleep states are defined in the System Sleep Control Register in the Power, Clocks and Resets Chapter

TABLE 33-13: ADDITIONAL VTR SUPPLY CURRENT WITH VARIOUS BLOCKS ENABLED

VCC	VTR	48 MHz Clock	VTR1			VTR_ANALOG			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On/Off	On	48MHz				0.45	0.50	0.50	mA	ADC enabled

Note 1: The values in this table are added to the values in VTR Supply Current, I_VTR excluding the sleep states.

TABLE 33-14: VBAT SUPPLY CURRENT, I_VBAT (VBAT=3.3V)

VCC	VTR	48 MHz PLL	Typical (3.3V, 25^0 C)	Max (3.3V, 25^0 C)	Units	Comments
Off	Off	Off	14	20	uA	Internal 32kHz oscillator - supplied by coin cell
Off	On	Off	65	70	uA	Internal 32kHz oscillator - add to VTR power well that supplies this current through the diode or is connected to the VBAT pin. This is not from the coin cell.
Off	Off	Off	14.5	20	uA	32kHz crystal oscillator
Off	Off	Off	14	20	uA	External 32kHz clock on XTAL2 pin -Running
Off	Off	Off	14	20	uA	External 32kHz clock on XTAL2 pin -Low

34.0 TIMING DIAGRAMS

34.1 Power-up and Power-down Timing

FIGURE 34-1: VTR/VBAT POWER-UP TIMING

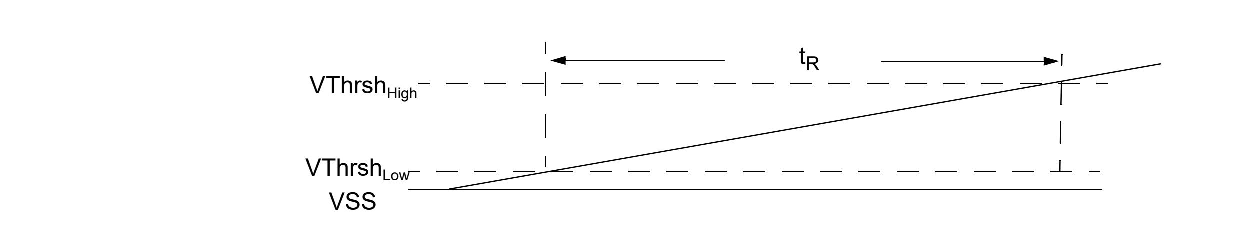


FIGURE 34-2: VTR RESET AND POWER-DOWN

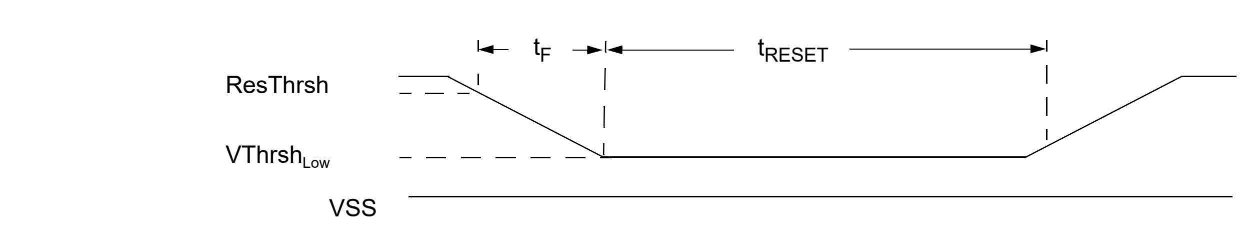


TABLE 34-1: VTR/VBAT TIMING PARAMETERS

Symbol	Parameter	MIN	TYP	MAX	Units	Notes
t_F	VTR Fall time	30			µs	1
	VBAT Fall time	30			µs	
	VTR_ANALOG Fall time	200			µs	
t_R	VTR Rise time	0.050		20	ms	1
	VBAT Rise time	0.100		20	ms	
	VTR_ANALOG Rise time	200			µs	
t_{RESET}	Minimum Reset Time	1			µs	
$VThrsh_{Low}$	VTR Low Voltage Threshold	0.1 × VTR			V	1
	VBAT Low Voltage Threshold	0.1 × VBAT			V	
$VThrsh_{High}$	VTR High Voltage Threshold			0.9 × VTR	V	1
	VBAT High Voltage Threshold			0.9 × VBAT	V	

Note 1: VTR applies to both VTR_REG and VTR_ANALOG

2: All values are guaranteed by design and not tested in production test

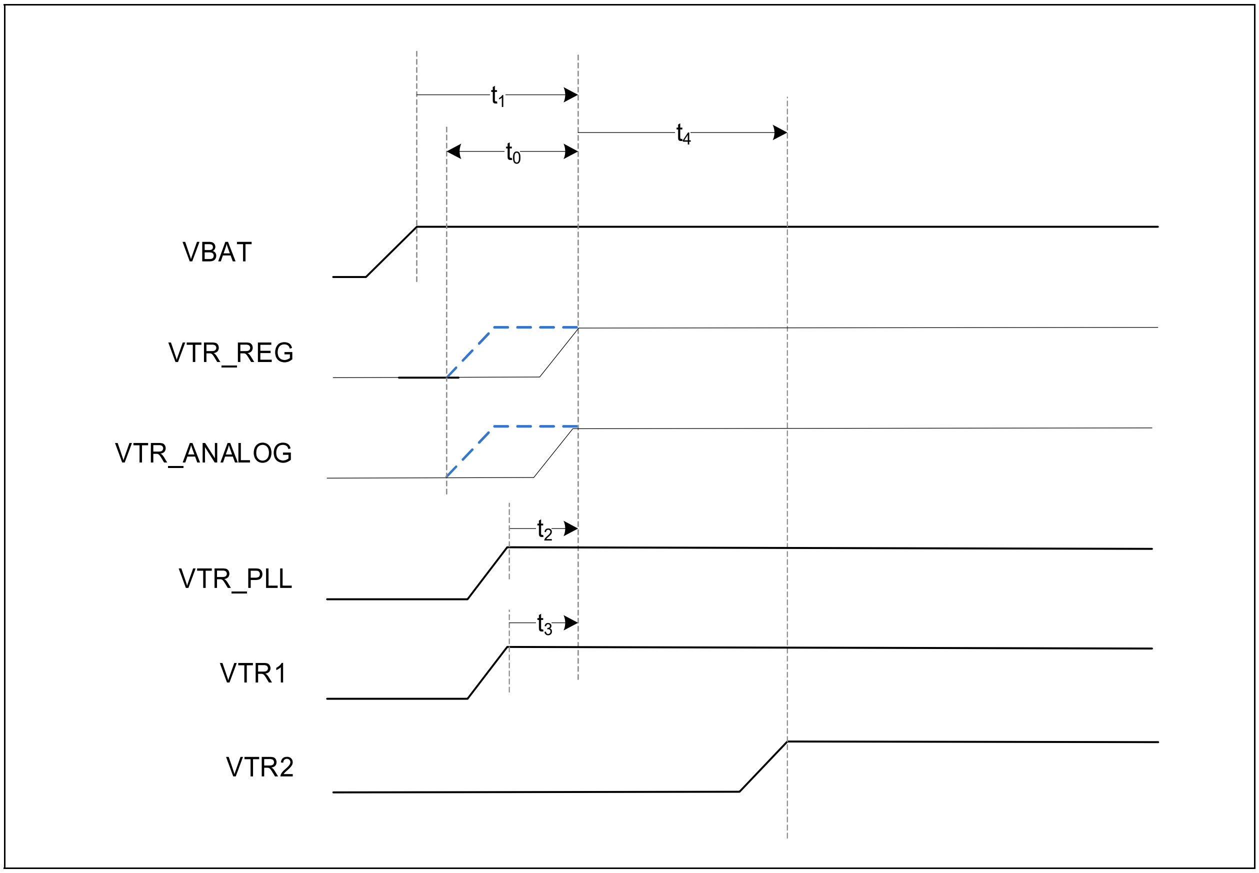
TABLE 34-1: VTR/VBAT TIMING PARAMETERS (CONTINUED)

Symbol	Parameter	MIN	TYP	MAX	Units	Notes
ResThrsh	VTR Reset Threshold	0.5	1.8	2.7	V	1
	VBAT Reset Threshold	0.5	1.25	1.9	V	

Note 1: VTR applies to both VTR_REG and VTR_ANALOG

2: All values are guaranteed by design and not tested in production test

34.2 Power Sequencing



Note 1: VTR_ANALOG and VTR_REG may ramp in either order

2: VBAT must rise no later than VTR_ANALOG and VTR_REG. This relationship is guaranteed by the recommended battery circuit in the layout guidelines.

3: The SHD_CS# pin, which is powered by VTR2, must be powered before the Boot ROM samples this pin.

4: Minimum operating threshold values for Power Rails are defined in Table 33-1, "Power Supply Operating Conditions".

34.3 nRESET_IN Timing

FIGURE 34-3: NRESET_IN TIMING

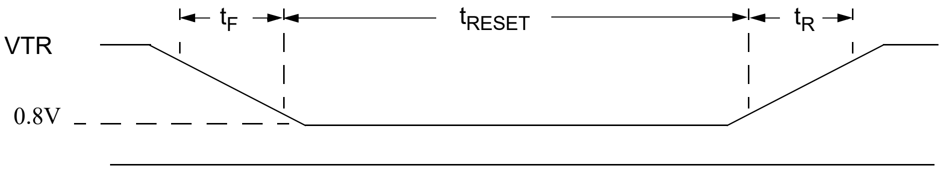


TABLE 34-2: RESETI# TIMING PARAMETERS

Symbol	Parameter	Limits		Units	Comments
		MIN	MAX		
t_F	nRESET_IN Fall time	0	1	ms	
t_R	nRESET_IN Rise time	0	1	ms	
t_{RESET}	Minimum Reset Time	1		µs	Note 1
Note 1: The nRESET_IN input pin can tolerate glitches of no more than 50ns.					

34.4 Clocking AC Timing Characteristics

FIGURE 34-4: CLOCK TIMING DIAGRAM

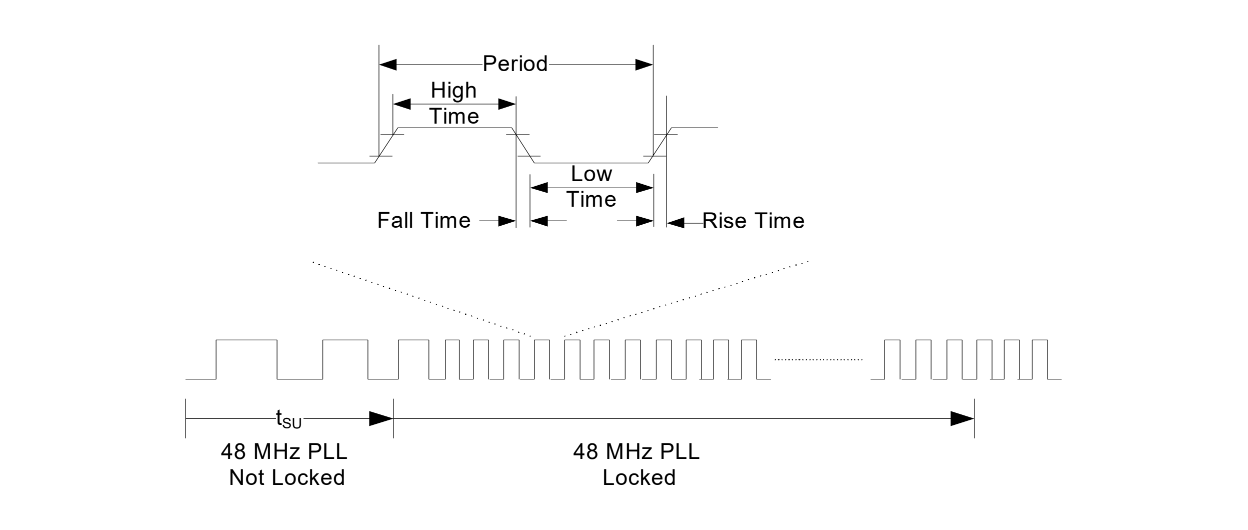


TABLE 34-3: CLOCK TIMING PARAMETERS

Clock	Symbol	Parameters	MIN	TYP	MAX	Units
48 MHz PLL	t_{SU}	Start-up accuracy from power-on-reset and waking from Heavy Sleep (Note 4)	-	-	3	ms
	-	Operating Frequency (locked to 32KHz single-ended input) (Note 1)	47.5	48	48.5	MHz
	-	Operating Frequency (Note 1)	46.56	48	49.44	MHz
	CCJ	Cycle to Cycle Jitter(Note 2)	-200		200	ps
	t_{DO}	Output Duty Cycle	45	-	55	%
32MHz Ring Oscillator	-	Operating Frequency(Note 5)	32	-	96	MHz

Note 1: The 48MHz PLL is frequency accuracy is computed by adding +/-1% to the accuracy of the 32kHz reference clock.

2: The Cycle to Cycle Jitter of the 48MHz PLL is +/-200ps based on an ideal 32kHz clock source. The actual jitter on the 48MHz clock generated is computed by adding the clock jitter of the 32kHz reference clock to the 48MHz PLL jitter (e.g., 32kHz jitter +/- 200ps).

3: An external single-ended 32KHz clock is required to have an accuracy of +/- 100 ppm.

4: PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset

5: The output of 32MHz Ring oscillator goes through a divide by 2 circuit to the 48MHz Core Clock

TABLE 34-3: CLOCK TIMING PARAMETERS (CONTINUED)

Clock	Symbol	Parameters	MIN	TYP	MAX	Units
32KHz single-ended input (Note 3)	-	Operating Frequency	-	32.768	-	kHz
	-	Period	(Note 3)	30.52	(Note 3)	µs
	-	High Time	10			us
	-	Low Time	10			us
	-	Fall Time	-	-	1	us
	-	Rise Time	-	-	1	us

Note 1: The 48MHz PLL is frequency accuracy is computed by adding +/-1% to the accuracy of the 32kHz reference clock.

2: The Cycle to Cycle Jitter of the 48MHz PLL is +/-200ps based on an ideal 32kHz clock source. The actual jitter on the 48MHz clock generated is computed by adding the clock jitter of the 32kHz reference clock to the 48MHz PLL jitter (e.g., 32kHz jitter +/- 200ps).

3: An external single-ended 32KHz clock is required to have an accuracy of +/- 100 ppm.

4: PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset

5: The output of 32MHz Ring oscillator goes through a divide by 2 circuit to the 48MHz Core Clock

34.5 GPIO Timings

FIGURE 34-5: GPIO TIMING

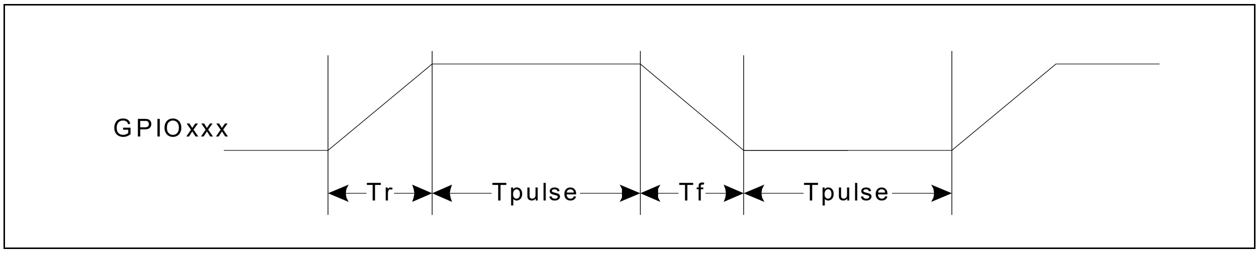


TABLE 34-4: GPIO TIMING PARAMETERS

Symbol	Parameter	MIN	TYP	MAX	Unit	Notes
t_R	GPIO Rise Time (push-pull)	0.54		1.31	ns	1
t_F	GPIO Fall Time	0.52		1.27	ns	
t_R	GPIO Rise Time (push-pull)	0.58		1.46	ns	2
t_F	GPIO Fall Time	0.62		1.48	ns	
t_R	GPIO Rise Time (push-pull)	0.80		2.00	ns	3
t_F	GPIO Fall Time	0.80		1.96	ns	
t_R	GPIO Rise Time (push-pull)	1.02		2.46	ns	4
t_F	GPIO Fall Time	1.07		2.51	ns	
t_{pulse}	GPIO Pulse Width	60			ns	

Note 1: Pad configured for 2ma, CL=2pF

2: Pad configured for 4ma, CL=5pF

3: Pad configured for 8ma, CL=10pF

4: Pad configured for 12ma, CL=20pF

34.6 Boot from SPI Flash Timing

Refer to CEC1712 Boot ROM document for the sequence and timing.

34.7 Serial Port (UART) Data Timing

FIGURE 34-6: SERIAL PORT DATA

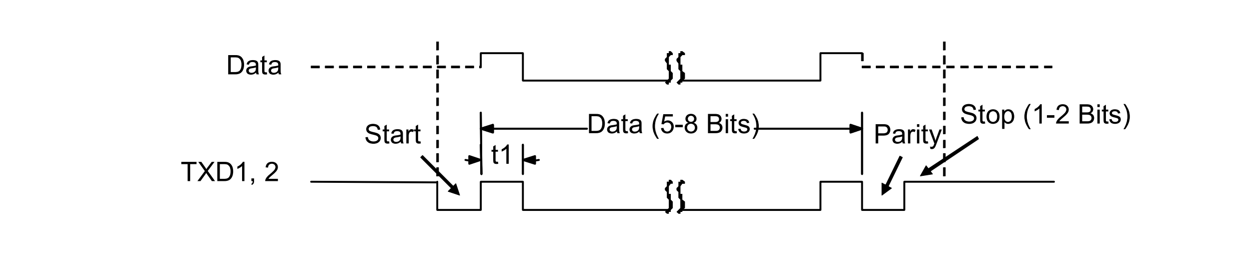


TABLE 34-5: SERIAL PORT DATA PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Serial Port Data Bit Time		t_{BR} (Note 1)		nsec

Note 1: tBR is 1/Baud Rate. The Baud Rate is programmed through the Baud_Rate_Divisor bits located in the Programmable Baud Rate Generator registers. The selectable baud rates are listed in Table 10-8, "UART Baud Rates using Clock Source 1.8432MHz" and Table 10-9, "UART Baud Rates using Clock Source 48MHz"Some of the baud rates have some percentage of error because the clock does not divide evenly. This error can be determined from the values in these baud rate tables.

34.8 PWM Timing

FIGURE 34-7: PWM OUTPUT TIMING

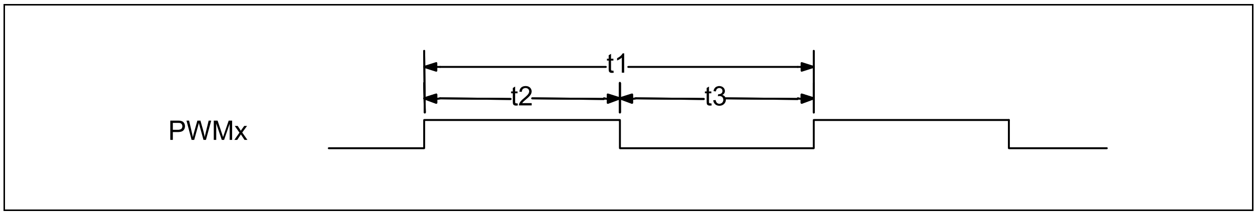


TABLE 34-6: PWM TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	42ns		23.3sec	
t_f	Frequency	0.04Hz		24MHz	
t2	High Time	0		11.65	sec
t3	Low Time	0		11.65	sec
t_d	Duty cycle	0		100	%

34.9 Fan Tachometer Timing

FIGURE 34-8: FAN TACHOMETER INPUT TIMING

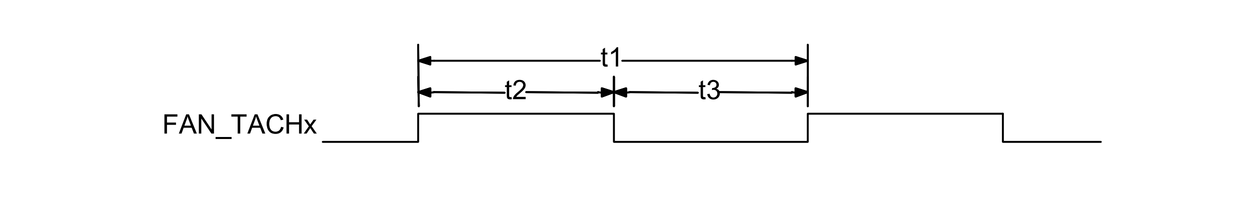


TABLE 34-7: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			μsec
t2	Pulse High Time	20			
t3	Pulse Low Time	20			
Note:	t_{TACH} is the clock used for the tachometer counter. It is 30.52 * prescaler, where the prescaler is programmed in the Fan Tachometer Timebase Prescaler register.				

34.10 Blinking/Breathing PWM Timing

FIGURE 34-9: BLINKING/BREATHING PWM OUTPUT TIMING

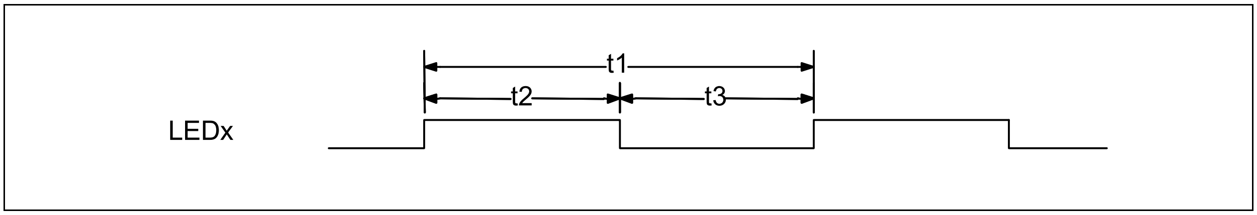


TABLE 34-8: BLINKING/BREATHING PWM TIMING PARAMETERS, BLINKING MODE

Name	Description	MIN	TYP	MAX	Units
t1	Period	7.8ms		32sec	
t_f	Frequency	0.03125		128	Hz
t2	High Time	0		16	sec
t3	Low Time	0		16	sec
t_d	Duty cycle	0		100	%

TABLE 34-9: BLINKING/BREATHING PWM TIMING PARAMETERS, GENERAL PURPOSE

Name	Description	MIN	TYP	MAX	Units
t1	Period	5.3µs		21.8ms	
t_f	Frequency	45.8Hz		187.5kHz	
t2	High Time	0		10.9	ms
t3	Low Time	0		10.9	ms
t_d	Duty cycle	0		100	%

34.11 I2C/SMBus Timing

FIGURE 34-10: I2C/SMBUS TIMING

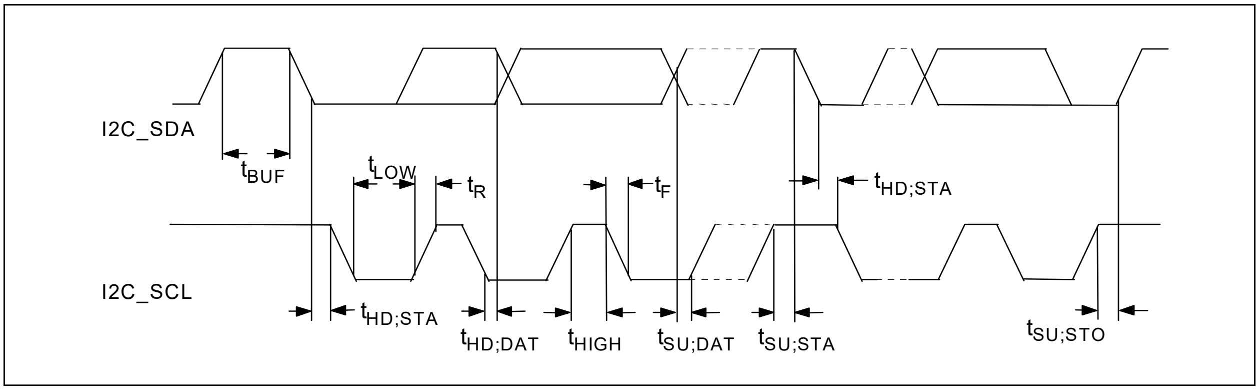


TABLE 34-10: I2C/SMBUS TIMING PARAMETERS

Symbol	Parameter	Standard-Mode		Fast-Mode		Fast-Mode Plus		Units
		MIN	MAX	MIN	MAX	MIN	MAX	
f_{SCL}	SCL Clock Frequency		100		400		1000	kHz
t_{BUF}	Bus Free Time	4.7		1.3		0.5		µs
$t_{SU;STA}$	START Condition Set-Up Time	4.7		0.6		0.26		µs
$t_{HD;STA}$	START Condition Hold Time	4.0		0.6		0.26		µs
t_{LOW}	SCL LOW Time	4.7		1.3		0.5		µs
t_{HIGH}	SCL HIGH Time	4.0		0.6		0.26		µs
t_R	SCL and SDA Rise Time		1.0		0.3		0.12	µs
t_F	SCL and SDA Fall Time		0.3		0.3		0.12	µs
$t_{SU;DAT}$	Data Set-Up Time	0.25		0.1		0.05		µs
$t_{HD;DAT}$	Data Hold Time	0		0		0		µs
$t_{SU;STO}$	STOP Condition Set-Up Time	4.0		0.6		0.26		µs

34.12 Quad SPI Master Controller - Serial Peripheral Interface (QMSPI) Timings

FIGURE 34-11: SPI CLOCK TIMING

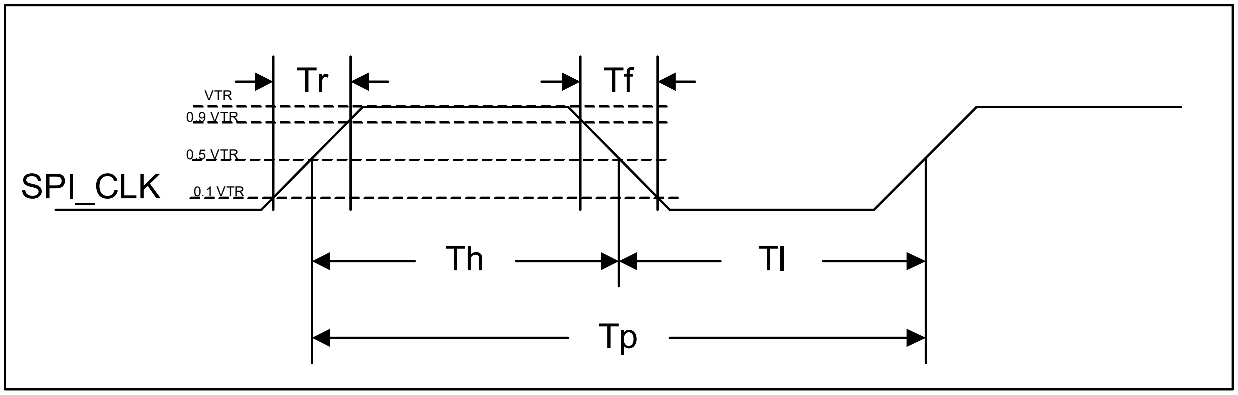


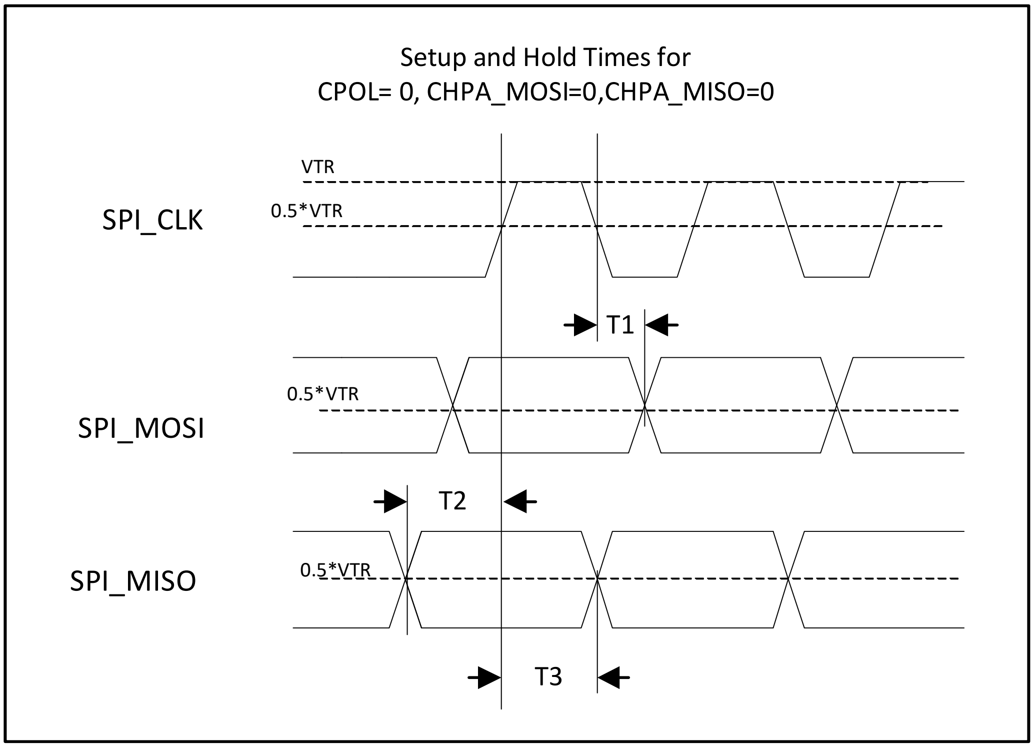
TABLE 34-11: SPI CLOCK TIMING PARAMETERS

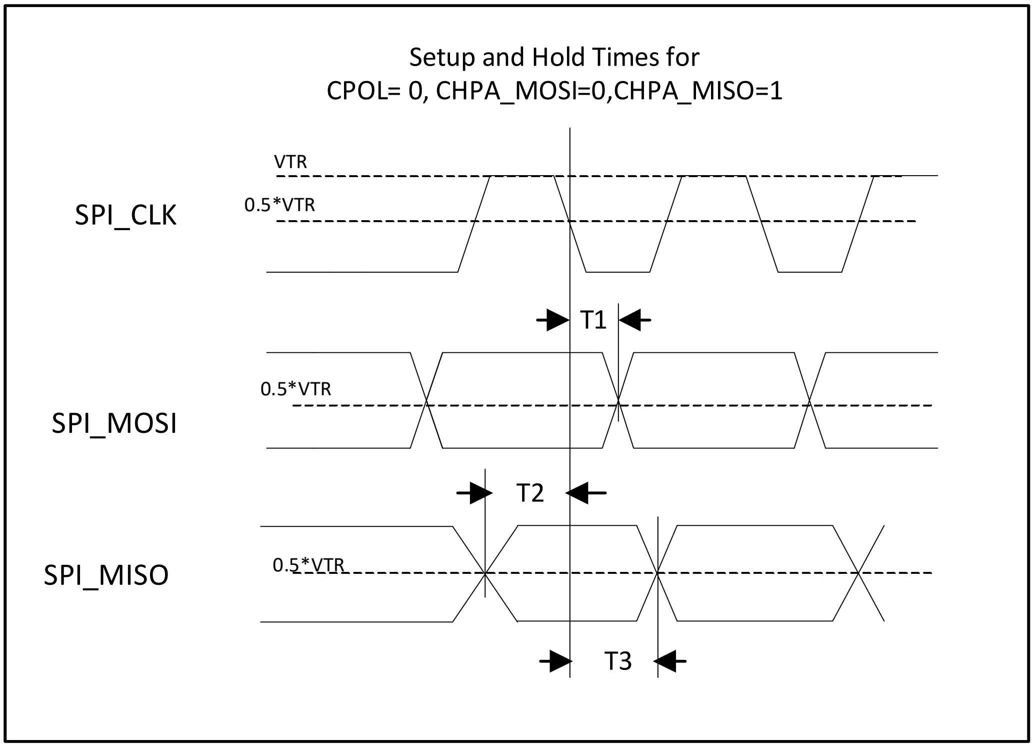
Name	Description	MIN	TYP	MAX	Units
Tr	SPI Clock Rise Time. Measured from 10% to 90%.			3	ns
Tf	SPI Clock Fall Time. Measured from 90% to 10%.			3	ns
Th/Tl	SPI Clock High Time/SPI Clock Low Time	40% of SPCLK Period	50% of SPCLK Period	60% of SPCLK Period	ns
Tp	SPI Clock Period – As selected by SPI Clock Generator Register	20.8		5,333	ns

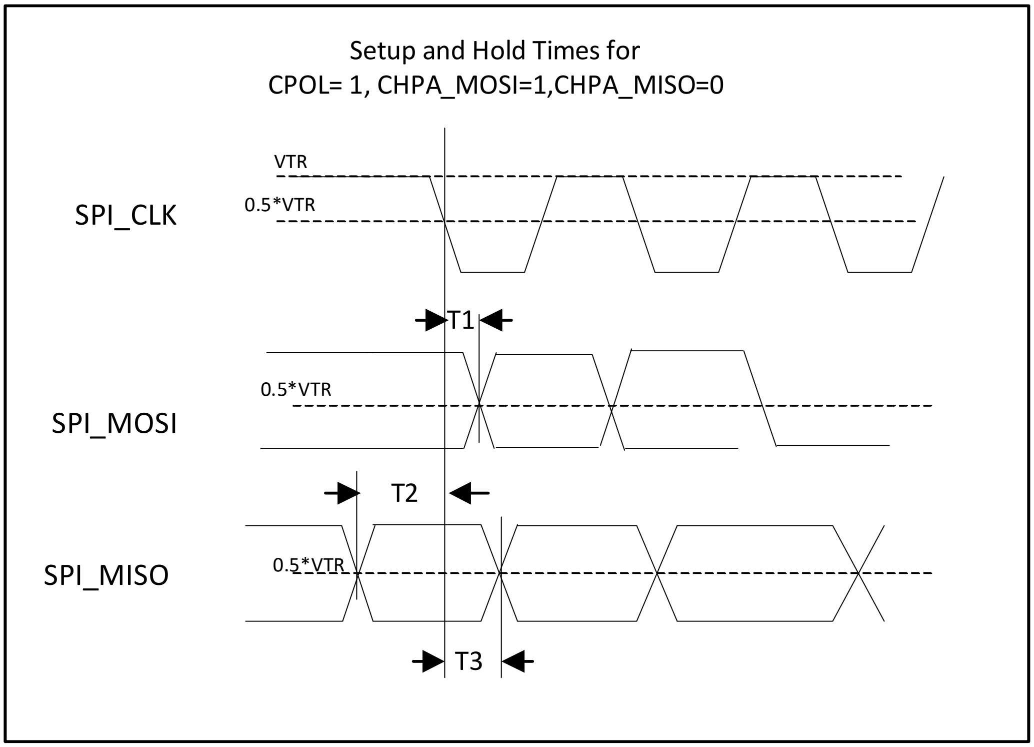
Note 1: Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow.

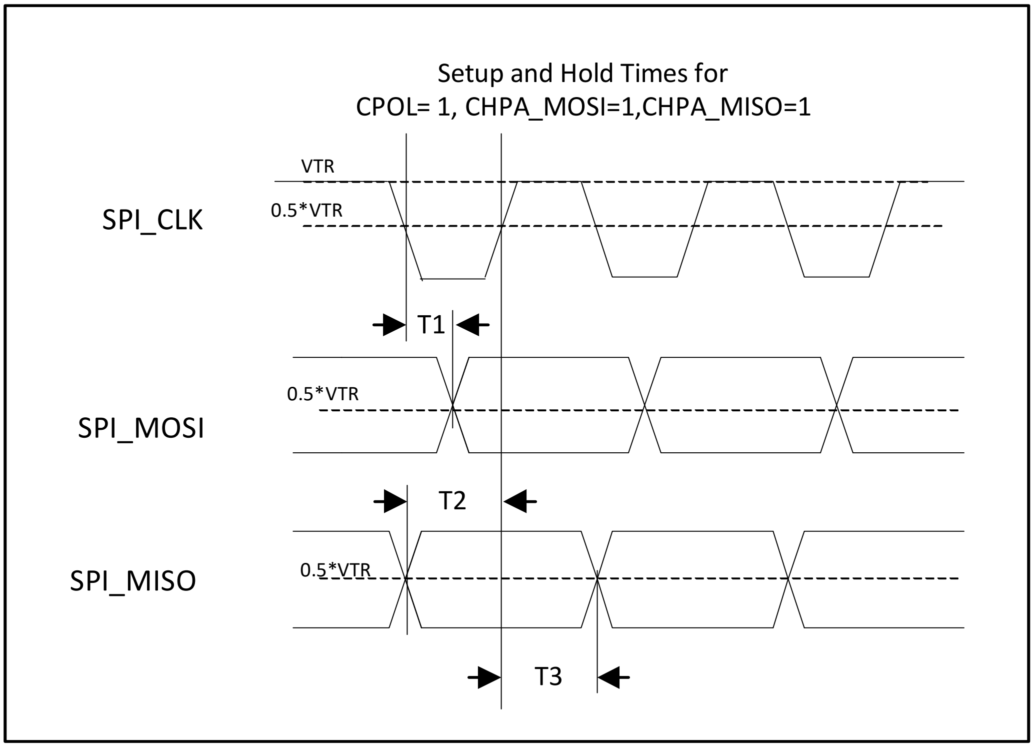
2: SPI_IO[3:0] signals also have the same Tr and Tf values

FIGURE 34-12: SPI SETUP AND HOLD TIMES






Setup and Hold Times for
CPOL= 1, CHPA_MOSI=1,CHPA_MISO=0
VTR
0.5*VTR
SPI_CLK
T1
0.5*VTR
SPI_MOSI
T2
0.5*VTR
SPI_MISO
T3


Setup and Hold Times for
CPOL= 1, CHPA_MOSI=1,CHPA_MISO=1
VTR
0.5*VTR
SPI_CLK
T1
0.5*VTR
SPI_MOSI
T2
0.5*VTR
SPI_MISO
T3

Note: SPI_IO[3:0] obey the SPI_MOSI and SPI_MISO timing. In the 2-pin SPI Interface implementation, SPI_IO0 pin is the SPI Master-Out/Slave-In (MOSI) pin and the SPI_IO1 pin is the Master-In/Slave-out (MISO) pin.

TABLE 34-12: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			2	ns
T2	Data IN Setup Time	0			ns
T3	Data IN Hold Time	5.1			ns

Note 1: Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow

2: Timing diagrams are not drawn to scale

34.13 Serial Debug Port Timing

FIGURE 34-13: SERIAL DEBUG PORT TIMING PARAMETERS

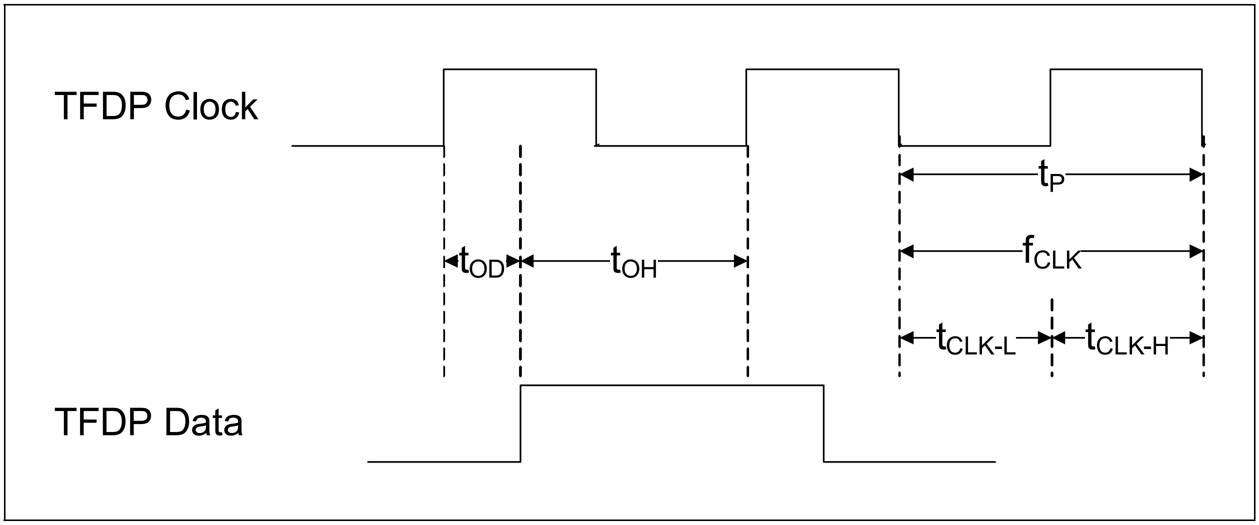


TABLE 34-13: SERIAL DEBUG PORT INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
f_{clk}	TFDP Clock frequency (see note)	2.5	-	24	MHz
t_P	TFDP Clock Period.	1/fclk			µs
t_{OD}	TFDP Data output delay after falling edge of TFDP_CLK.			5	nsec
t_{OH}	TFDP Data hold time after falling edge of TFDP Clock	t_P - t_{OD}			nsec
t_{CLK-L}	TFDP Clock Low Time	t_P/2 - 3		t_P/2 + 3	nsec
t_{CLK-H}	TFDP Clock high Time (see Note 1)	t_P/2 - 3		t_P/2 + 3	nsec

Note 1: When the clock divider for the embedded controller is an odd number value greater than 2h, then t_{CLK-L} = t_{CLK-H} + 15 ns. When the clock divider for the embedded controller is 0h, 1h, or an even number value greater than 2h, then t_{CLK-L} = t_{CLK-H}.

34.14 JTAG Interface Timing

FIGURE 34-14: JTAG POWER-UP & ASYNCHRONOUS RESET TIMING

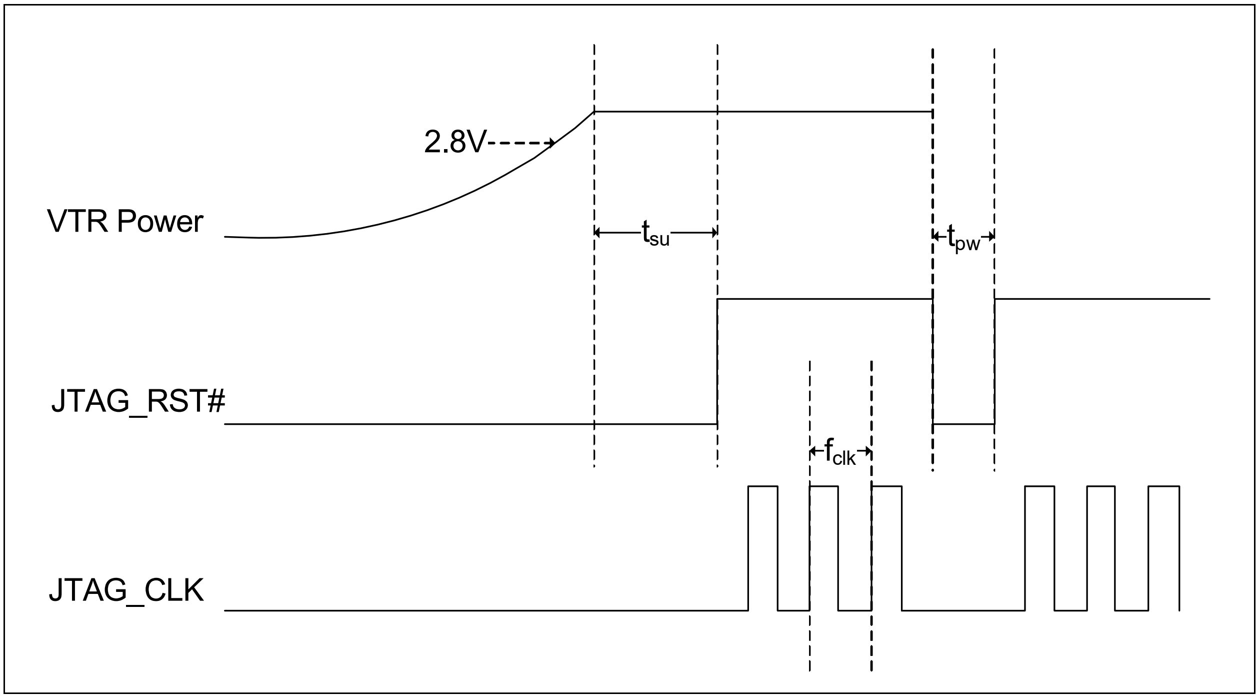


FIGURE 34-15: JTAG SETUP & HOLD PARAMETERS

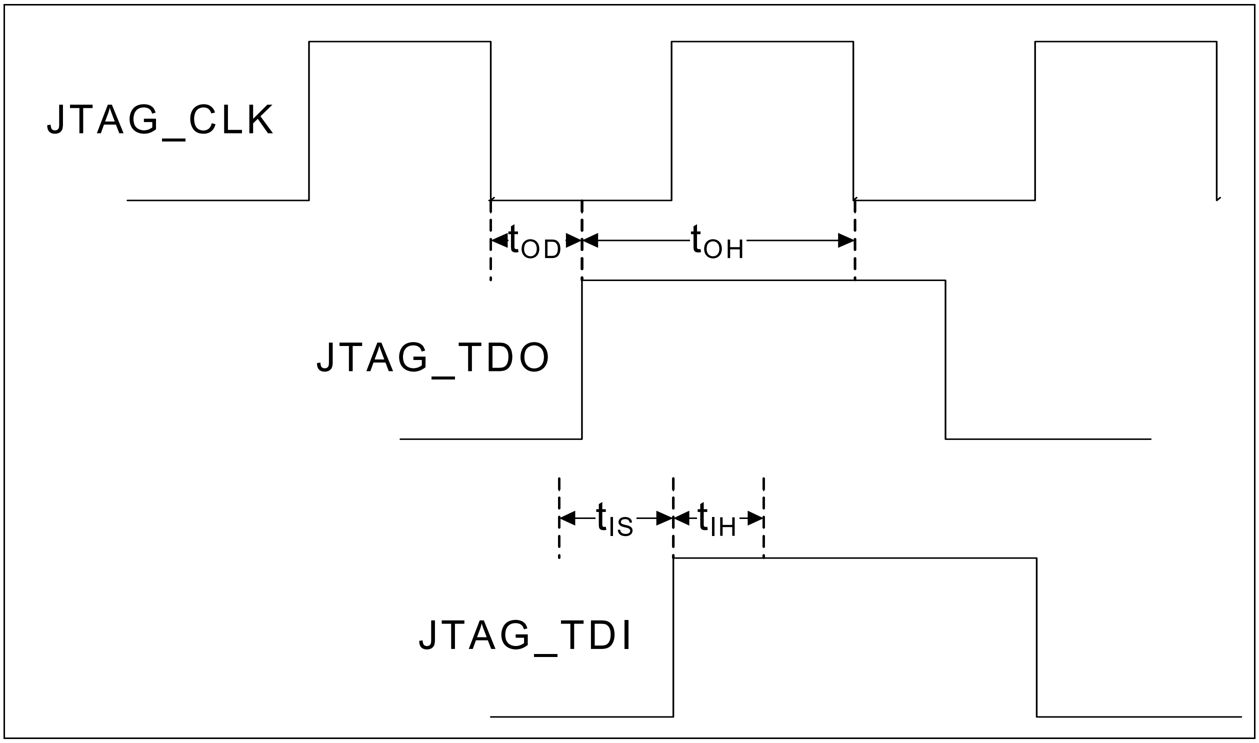


TABLE 34-14: JTAG INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{su}	JTAG_RST# de-assertion after VTR power is applied	5			ms
t_{pw}	JTAG_RST# assertion pulse width	500			nsec
f_{clk}	JTAG_CLK frequency (see note)			48	MHz
t_{OD}	TDO output delay after falling edge of TCLK.	5		10	nsec
t_{OH}	TDO hold time after falling edge of TCLK	1 TCLK - t_{OD}			nsec
t_{IS}	TDI setup time before rising edge of TCLK.	5			nsec
t_{IH}	TDI hold time after rising edge of TCLK.	5			nsec

Note: f_{clk} is the maximum frequency to access a JTAG Register.

APPENDIX A: DATA SHEET REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision	Section/Figure/Entry	Correction
DS00003416D (03-18-22)	Section 2.7, "Package Information"	SX1 package replaced with 2ZW.
	Table 3-5, "Register Map"	Register address for ARMM4 changed from “DFFF” to “E000”.
	Section 12.7.5, "WDT Status Register" and Section 12.7.6, "WDT Int Enable Register"	Reset event changed from “RESET_SYS_nWDT” to “RESET_SYS”.
	Section 21.8.3, "Application Notes"	Added additional notes.
	Product Identification System	“SX1” replaced with new package “2ZW” Added: “S” = Soteria Version
DS00003416C (07-01-21)	Section TABLE 4-4:, "Source Clock Definitions"	Edited 32Mhz Ring oscillator description.
	Section 33.2.4.1, "Pin Leakage"	Added Pin leakage current values from testing.
	Section 34.12, "Quad SPI Master Controller - Serial Peripheral Interface (QMSPI) Timings"	Added Timing diagrams for Mode0 -3. Updated Timing Setup and Hold time from testing.
DS00003416B (10-21-20)	Figure 4-3, "Resets Block Diagram"	Diagram added.
	Table 4-4, "Source Clock Definitions"	Table updated with crystal oscillator information.
	Section 4.5.4, "32KHz Clock"	Section updated with crystal oscillator information.
	Table 33-14, "VBAT Supply Current, I_VBAT (VBAT=3.3V)"	Table added.
	Section 34.12, "Quad SPI Master Controller - Serial Peripheral Interface (QMSPI) Timings"	Section added.
DS00003416A (03-05-20)	Document Release	

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.[1]	-	**X**	-	**XX**	-	**XXX**[2]	-	**[X]**[3]
Device		**Total SRAM**		**Version/ Revision**		**Temp range/ Package**		**Tape &Reel Option**

Device:	CEC1712[1]	Embedded Controller with Thermal Monitor Integration
Total SRAM	H	256KB

Version/ Revision:	B#
	S#
		B = Standard Version, # = Version Revision Number
		S = Soteria Version, # = Version Revision Number

Temperature Range	I/	-40°C to +85°C (Industrial)
Package:	2ZW	84 pin WFBGA[2]), 7mm x 7mm body, 0.65mm pitch

Tape and Reel Option:	Blank = Tray packaging
	TR = Tape and Reel[3]	

Example:

a) CEC1712H-B2-I/2ZW = CEC1712, 256KB total SRAM, Standard ROM, ROM Version 2, 84-WFBGA, Tray packaging

Note 1: These products meet the halogen maximum concentration values per IEC61249-2-21.

2: All package options are RoHS compliant. For RoHS compliance and environmental information, please visit http://www.microchip.com/pagehandler/en-us/aboutus/ehs.html .

3: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at https://www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WAR- RANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON- INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDI- RECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, Symmcom, and Trusted Time are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 9781668300268

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

MICROCHIP

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra’anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820