EEC1727

32-Bit Embedded Controller with Secure Boot & Crypto Hardware

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-M4F Embedded Processor

- Programmable clock frequency up to 96 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

- 416KB Code/Data SRAM
  - 352KB optimized for code performance
  - 64KB optimized for data performance
- 128 Bytes Battery Powered Storage SRAM
- 4K bits OTP
  - In circuit programmable
- ROM
- Contains Boot ROM
  - Contains Runtime APIs for built-in functions
- 128KB of ROM space
- 4Mbit (512KByte) in-chip SPI Serial Flash in specific packages (Refer Internal SPI in Table 1-1)
  - SST25PF040C
  - SPI Master controller
  - Supports Mode 0 and Mode 3
  - 24MHz

Clocks

- 96 MHz Internal PLL
- 32 kHz Clock Sources
  - Internal 32 kHz silicon oscillator

Package Options

- 68 pin WFBGA

Security Features

- Boot ROM Secure Boot Loader
  - Hardware Root of Trust (RoT) using Secure Boot and Immutable code
  - 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
- 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) Slave
  - Quad SPI (half-duplex) or Single wire (full duplex) support
  - Mode 0 and Mode3 operation
  - Programmable wait time for response delay

© 2021 Microchip Technology Inc.

- 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 the following ports (only 1 port active at a time)
    - 1 shared SPI Interface1 Private SPI Interface
    - 1 In-Chip SPI
- Internal DMA Controller
  - Hardware or Firmware Flow Control
  - Firmware Initiated Memory-to-Memory transfers
  - Hardware CRC-32 Generator on Channel 0
  - 16-Hardware DMA Channels support five SMBus Master/Slave Controllers, One Quad SPI Controller and Two General purpose SPI Controllers
- I2C/SMBus Controllers
  - 5 I2C/SMBus controllers
  - 6 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
  - 8 GPIO Pass through ports
- Input Capture and Compare timer
  - Six 32-bit Capture Registers
  - 5 Input Pins (ICTx)
    - Full Crossbar switch allows any port to be connected to any controller
  - 32-bit Free-running timer
  - Two 32-bit Compare Registers
  - Capture, Compare and Overflow Interrupts
- Universal Asynchronous Receiver Transmitter (UART)
  - Two High Speed NS16C550A Compatible UARTs with Send/Receive 16-Byte FIFOs
    - UART0 - 2-Pin
    - UART1 - 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)
  - Watchdog reset IRQ vector
- 4 Programmable Pulse Width Modulator (PWM) outputs
  - Multiple Clock Rates
  - 16-Bit ON & 16-Bit OFF Counters
- 2 Fan Tachometer Inputs
  - 16 Bit Resolution
- Two RPM-Based Fan Speed Controllers
  - Each includes one Tach input and one PWM output
  - Each includes one Tach input and one PWM output
  - 3% accurate from 500 RPM to 16k RPM
  - Automatic Tachometer feedback
  - Aging Fan or Invalid Drive Detection

© 2021 Microchip Technology Inc.

  - Spin Up Routine
  - Ramp Rate Control
  - RPM based Fan Control Algorithm
- Breathing LED Interface
  - 2 Blinking/Breathing LEDs
  - Programmable Blink Rates
  - Piecewise Linear Breathing LED Output Controller
    - Provides for programmable rise and fall waveforms
  - Operational in EC Sleep States
- Optional support for Physically Unclonable Function (PUF)
  - 2K Bit memory reserved for PUF.

Analog Features

- ADC Interface
  - 10-bit or 12-bit readings supported
  - ADC Conversion time 500nS/channel
  - 8 Channels
  - External voltage reference
  - Supports thermistor temperature readings
- Two Analog Comparators
  - May be used for Hardware Shutdown
    - Detection of voltage limit event
    - Detection of Thermistor Over-Temp Event

Battery Powered Peripherals

- Real Time Clock (RTC)
  - VBAT Powered
  - 32KHz Silicon Oscillator
  - 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

Debug Features

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

© 2021 Microchip Technology Inc.

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.

© 2021 Microchip Technology Inc.

Table of Contents

1.0 GENERAL DESCRIPTION

The EEC1727 is a low power integrated embedded controller designed for security and storage enclosure platforms. The EEC1727 is a highly-configurable, mixed-signal, advanced I/O controller. It contains a 32-bit ARM® Cortex-M4F 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 bootloader 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 EEC1727 device is directly powered by a minimum of two separate suspend supply planes (VBAT and VTR) and senses a third runtime power plane (VCC) to provide "instant on" and system power management functions. The EEC1727 has one banks of I/O pins that are able to operate at 3.3 V (VTR1), one bank that is 1.8V (VTR3) and one bank that can operate at 3.3V/1.8V (VTR2). Operating at 1.8V allows the EEC1727 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 EEC1727 secure bootloader authenticates and optionally decrypts the SPI Flash OEM boot image using the AES-256, ECDSA, SHA-512 cryptographic hardware accelerators. The EEC1727 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.

The EEC1727 is designed to be incorporated into low power PC architecture designs and supports ACPI sleep states (S0-S5). During normal operation, the hardware always operates in the lowest power state for a given configuration. When the chip is sleeping, it has many wake events that can be configured to return the device to normal operation. Some examples of supported wake events are PS2 wake events, RTC, Week Alarm, Hibernation Timer, or any GPIO pin.

The EEC1727 offers a software development system interface that includes a Trace FIFO Debug port, a host accessible serial debug port with a 16C550A register interface, a Port 80 BIOS Debug Port, and a 2-pin Serial Wire Debug (SWD) interface. Also included is a 4-wire JTAG interface used for Boundary Scan testing.

Note: 4-wire JTAG interface can only be used for Boundary Scan testing.

1.1 Family Features

TABLE 1-1: EEC1727 FEATURE LIST

EEC1727 Product Features	EEC1727
Device ID	0x002217XX
JTAG ID	0x02242445
Package	68 WFBGA
Total SRAM Options	416KB
Code/Data Options (Primary Use)	352KB/64KB
Battery Backed SRAM	128 bytes
EEPROM Controller Supports 2KB	No
Internal SPI	Yes
XiP CACHE	Yes
2 pin SWD	Yes
4 pin JTAG Boundary Scan Only	Yes
RPMC	Yes
SPI Slave	Yes

© 2021 Microchip Technology Inc.

EEC1727 Product Features	EEC1727
Embedded Memory Interface (EMI)	3
Mailbox Register Interface	1
ACPI Embedded Memory Controller Interface	No
ACPI PM1 Block Interface	No
eSPI	No
Trace FIFO Debug Port	Yes
Internal DMA Channels	16
16-bit Basic Timer	4
32-bit Basic Timer	2
16-bit Counter/Timer	4
Capture Timer	2
ICT Channels	5
Compare Timer	1
Watchdog Timer (WDT)	1
Hibernation Timer	2
Week Timer	1
Sub Week Timer	1
RTC	1
RTOS Timer	1
SMBus Network 2.0/ I2C Master Controllers	5
SMBus Ports	6
GPIOs	55
Blinking/Breathing LED	2
Quad SPI Master Controller	1 controller/ 2 ports
10/12-bit ADC Channels	8
Vref-2 ADC	Yes
RPM2PWM	2
16-bit PWMs	6
16-bit TACHs	2
UARTs	2 UART0: 2-pin UART1: 2-pin
AES Hardware Support	128-256 bit
SHA 1, SHA 2 and Hashing Support	SHA-1 to SHA-2
Public Key Cryptography Support	RSA: 4K bit ECC: 571 bit
True Random Number Generator with health test	1K bit
User OTP	4K bits
Analog Comparator	Yes
5V Tolerant Pads	6
GPIO Pass Through Ports (GPTP)	8
BC-Link	No

© 2021 Microchip Technology Inc.

EEC1727 Product Features	EEC1727
RC-ID	No
Differential Power Analysis counter-measures (DPA)	Yes
Root Of Trust	Yes
Secure Boot	Yes
Immutable Code	Yes
Key Revocation	Yes
Key Roll Back Protection	127
Optional PUF support	Yes

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

1.2 Boot ROM

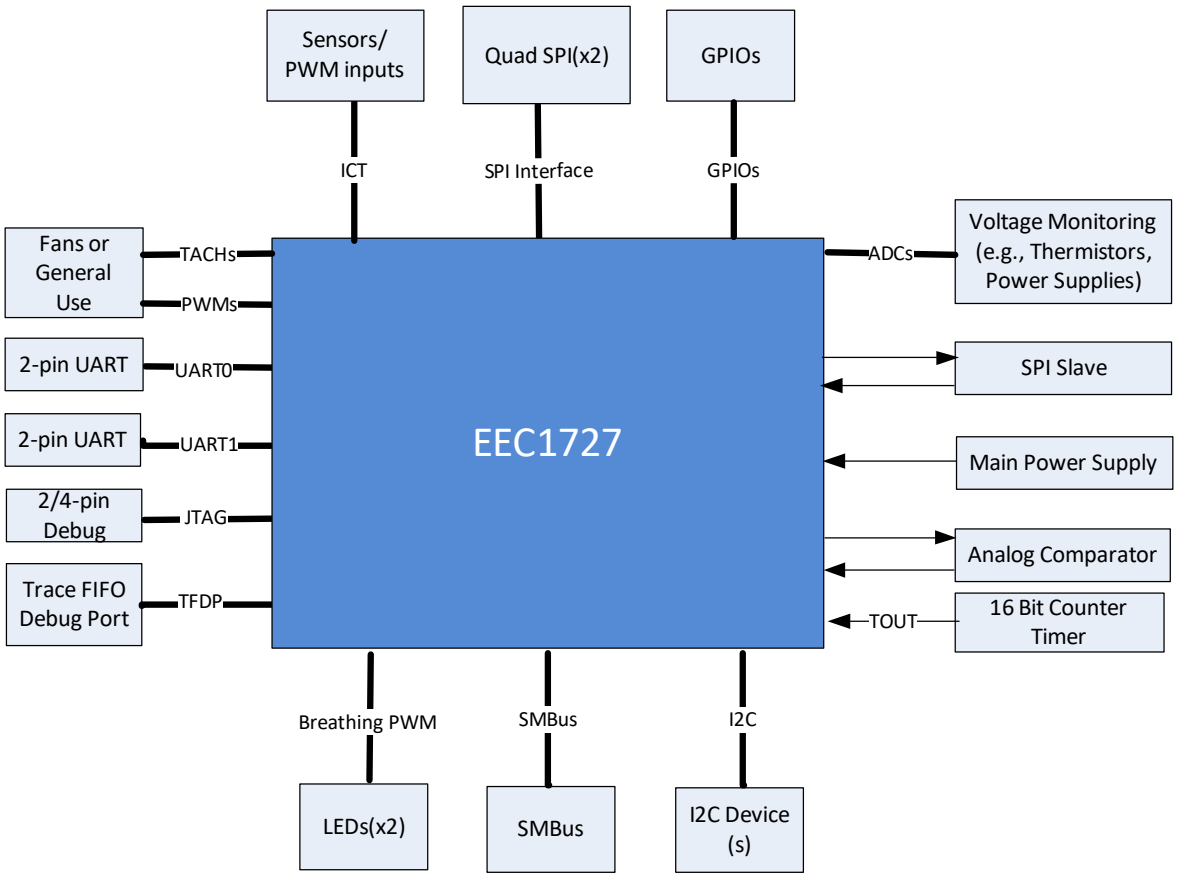
Following the release of the RESET_EC signal, the processor will start executing code from the Boot ROM. The Boot ROM executes the SPI Flash Loader, which downloads User Code from SPI Flash and stores it in the internal Code RAM. Refer to EEC1727 Boot ROM document for further details.

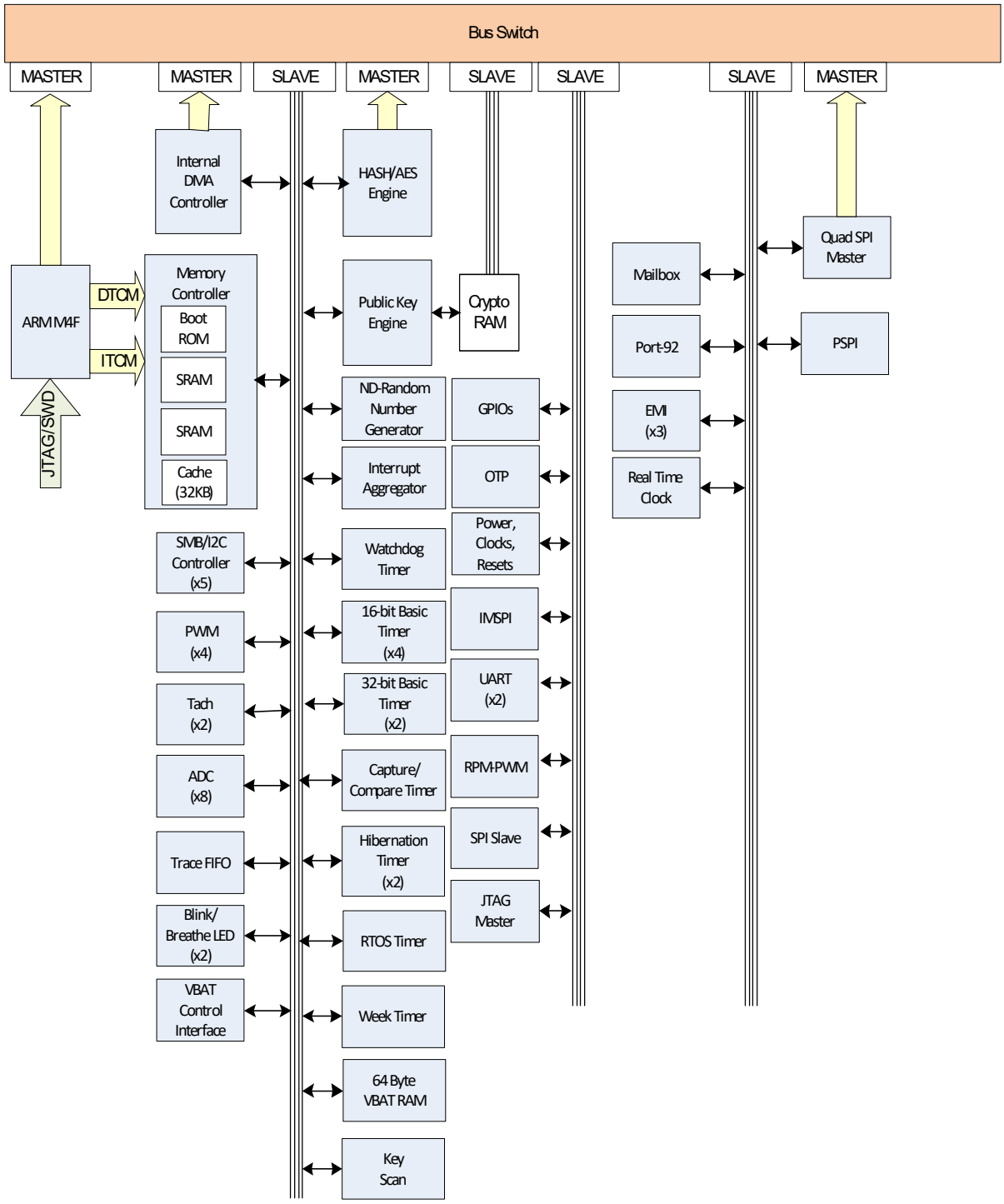
1.3 EEC1727 Internal Address Spaces

The Internal Embedded Controller can access any register in the EC Address Space or Host Address Space. If the I^2C interface is used as the Host Interface, access to all the IP Peripherals is dependent on EC firmware.

© 2021 Microchip Technology Inc.

FIGURE 1-1: BLOCK DIAGRAM





© 2021 Microchip Technology Inc.

2.0 PIN CONFIGURATION

2.1 Description

The Pin Configuration chapter includes Pin List, Pin Multiplexing and Package Information.

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. GPIO pin drive strength is determined by the Pin Control Register Defaults field in the Pin Control Register 2. **Note:** In the Table 2-2, "EEC1727 68 WFBGA PIN MUX TABLE" these are represented as PIO with empty drive strength column for that row and in Table 37-3, "DC Electrical Characteristics" these are represented as PIO-12.
In	I Type Input Buffer.
O2	O-2 mA Type Buffer.
SB-TSI	SB-TSI Input/Output. These pins operate at the processor voltage level (VREF_VTT)
High Drive Pad	Configurable drive strength of 4,8,16,24 mA. In the Table 2-2, "EEC1727 68 WFBGA PIN MUX TABLE" these are represented as PIO with 24mA drive strength column for that row and in Table 37-3, "DC Electrical Characteristics" these are represented as PIO-24.

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. For example, GPIO062(RESETO#) has only a single mux option, GPIO062, but the signal GPIO062 can also be used or interpreted as RESETO#.
- 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: EEC1727 68 WFBGA PINOUT

Pin Map	Signal
G1	GPIO005/I2C11_SDA_ALT/GPTP_OUT4
G2	GPIO006/I2C11_SCL_ALT/GPTP_OUT7
B1	GPIO007/I2C03_SDA
C1	GPIO010/I2C03_SCL
G11	GPIO012/I2C07_SDA/SLV_SPI_IO2/TOUT3

TABLE 2-1: EEC1727 68 WFBGA PINOUT (CONTINUED)

Pin Map	Signal
G10	GPIO013/I2C07_SCL/SLV_SPI_IO3/TOUT2
F8	GPIO014/PWM6/SLV_SPI_IO1/GPTP_IN2
H10	GPIO016/GPTP_IN1/ICT3
C11	GPIO017/GPTP_IN5
K1	GPIO022/GPTP_IN4/32kHz_OUT_ALT
H2	GPIO023/GPTP_IN7
J2	GPIO024/GPTP_IN6/I2C07_SCL_ALT
B10	GPIO031/GPTP_OUT1
B11	GPIO032/GPTP_OUT0
B9	GPIO035/PWM8/CTOUT1/ICT15/LED3
C10	GPIO040/GPTP_OUT2
K9	GPIO050/ICT0_TACH0/GTACH0
K10	GPIO051/ICT1_TACH1/GTACH1
E8	GPIO053/PWM0/SLV_SPI_MSTR_INT/GPWM0
D8	GPIO054/PWM1/SLV_SPI_SCLK/GPWM1
L3	GPIO057/CMP_VIN0
K11	GPIO067/VREF2_ADC
A9	GPIO104/UART0_TX/TFDP_CLK_ALT
A10	GPIO105/UART0_RX/TFDP_DATA_ALT
H8	GPIO106/CMP_VREF1
A5	GPIO121/PVT_IO0
B5	GPIO122/PVT_IO1
B4	GPIO123/PVT_IO2
B3	GPIO124/PVT_CS#/ICT12/GPTP_OUT6
A2	GPIO125/PVT_CLK/GPTP_OUT5
A3	GPIO126/PVT_IO3[UART_BSTRAP]
E11	GPIO130/I2C01_SDA/SLV_SPI_IO0/TOUT1
E10	GPIO131/I2C01_SCL/SLV_SPI_CS#/TOUT0
D5	GPIO145/I2C09_SDA/JM_TDI
D6	GPIO146/I2C09_SCL/ITM/JM_TDO/JTAG_TDO
E1	GPIO147/I2C15_SDA/JM_TCLK
E2	GPIO150/I2C15_SCL/JM_TMS//JTAG_TMS
D10	GPIO152/GPTP_OUT3/I2C07_SDA_ALT
C2	GPIO156/LED0
D4	GPIO165/32KHZ_IN/CTOUT0
B8	GPIO170/UART1_TX/TFDP_CLK[JTAG_STRAP]
B7	GPIO171/UART1_RX/TFDP_DATA
B2	GPIO175/CMP_VOUT1/PWM8_ALT
K4	GPIO200/ADC00
L10	GPIO201/ADC01
L9	GPIO202/ADC02
K5	GPIO203/ADC03
H6	GPIO204/ADC04
L7	GPIO205/ADC05

© 2021 Microchip Technology Inc.

TABLE 2-1: EEC1727 68 WFBGA PINOUT (CONTINUED)

Pin Map	Signal
K7	GPIO206/ADC06
K8	GPIO207/ADC07[CMP_STRAP]
K3	GPIO221/32KHz_OUT/GPTP_IN3/CMP_VIN1
J11	GPIO224/GPTP_IN0
H5	GPIO226/CMP_VREF0
D2	GPIO241/CMP_VOUT0/PWM0_ALT
A7	JTAG_RST#
K2	nRESET_IN
L5	VR_CAP
J10	VREF_ADC
F4	VSS
G4	VSS
H7	VSS_ADC
J1	VSS_ANALOG
D7	VTR_ANALOG
L2	VTR_PLL
H4	VTR_REG
E4	VTR1
G8	VTR2

Note: VBAT defined logic in the device is connected to the VTR1 power well.

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 Table 3-4, "GPIO Pin Control Default Values".

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 “O”, But 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 37.0, "Electrical Specifications".

- I/O Pad Types are defined in Section 37.2.4, "DC Electrical Characteristics for I/O Buffers".
- The abbreviation “PWR” is used to denote power pins. The power supplies are defined in Section 37.2.1, "Power Supply Operational Characteristics".

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 (OVP)

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 (BDP)

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.

© 2021 Microchip Technology Inc.

2.4.10 PIN MULTIPLEXING

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
Default :0	GPIO005	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	I2C11_SDA_ALT	PIO			All PGS options	High			
2	GPTP_OUT4	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO006	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	I2C11_SCL_ALT	PIO			All PGS options	High			
2	GPTP_OUT7	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO007	PIO		VTR1	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	I2C03_SDA	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO010	PIO		VTR1	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	I2C03_SCL	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
Default :0	GPIO012	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C07_SDA	PIO			All PGS options	High			
2	SLV_SPI_IO2	PIO			All PGS options	Low			
3	TOUT3	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO013	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C07_SCL	PIO			All PGS options	High			
2	SLV_SPI_IO3	PIO			All PGS options	Low			
3	TOUT2	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO014	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	PWM6	O			All PGS options	NA			
2	SLV_SPI_IO1	PIO			All PGS options	Low			
3	GPTP_IN2	I			All PGS options	Low			
4	Reserved								
5	Reserved								
Default :0	GPIO016	PIO-24	24mA	VTR2	All PGS options	No Gate	Yes	Yes	Drive strength can be configured by Pin Control register2 to 4,8,16 or 24mA
1	GPTP_IN1	I			All PGS options	Low			
2	Reserved								
3	ICT3	I			All PGS options	Low			
4	Reserved								
5	Reserved								
Default :0	GPIO017	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	Reserved								
3	GPTP_IN5	I			All PGS options	Low			

© 2021 Microchip Technology Inc.

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
4	Reserved								
5	Reserved								
Default :0	GPIO022	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	Reserved								
3	GPTP_IN4	I			All PGS options	Low			
4	32kHz_OUT_ALT	O			All PGS options	NA			
5	Reserved								
Default :0	GPIO023	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	GPTP_IN7	I			All PGS options	Low			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO024	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	GPTP_IN6	I			All PGS options	Low			
3	I2C07_SCL_ALT	PIO			All PGS options	High			
4	Reserved								
5	Reserved								
Default :0	GPIO031	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	GPTP_OUT1	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO032	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	GPTP_OUT0	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
Default :0	GPIO035	PIO		VTR1	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	PWM8	O			All PGS options	NA			
2	CTOUT1	O			All PGS options	NA			
3	ICT15	I			All PGS options	Low			
4	LED3	O			All PGS options	NA			
5	Reserved								
Default :0	GPIO040	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	GPTP_OUT2	O			All PGS options	NA			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO050	PIO		VTR1	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	ICT0_TACH0	I			All PGS options	Low			
2	GTACH0	I			All PGS options	Low			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO051	PIO		VTR1	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	ICT1_TACH1	I			All PGS options	Low			
2	GTACH1	I			All PGS options	Low			

© 2021 Microchip Technology Inc.

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO053	PIO		VTR2	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	PWM0	O			All PGS options	NA			
2	SLV_SPI_MSTR_INT	O			All PGS options	NA			
3	GPWM0	O			All PGS options	Low			
4	Reserved								
5	Reserved								
Default :0	GPIO054	PIO		VTR2	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	PWM1	O			All PGS options	Low			
2	SLV_SPI_SCLK	I			All PGS options	Low			
3	GPWM1	O			All PGS options	Low			
4	Reserved								
5	Reserved								
Default :0	GPIO057	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	Reserved								
2	CMP_VIN0	I_AN			PGS=00 (only)	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO067	PIO		VTR1	All PGS options	No Gate	Yes	Yes	

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
1	VREF2_ADC	O			PGS=00 (only)	Low			GPIO067/VREF2_ADC used as a GPIO can inject noise into the ADC. Hence care should be taken in system
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO104	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	UART0_TX	O			All PGS options	NA			
2	TFDP_CLK_ALT	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO105	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	UART0_RX	I			All PGS options	Low			
2	TFDP_DATA_ALT	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO106	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	Reserved								
2	CMP_VREF1	I_AN			PGS=00 (only)	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO121	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	PVT_IO0	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO122	PIO		VTR1	All PGS options	No Gate	Yes	Yes	

© 2021 Microchip Technology Inc.

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
1	PVT_IO1	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	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								
4	Reserved								
5	Reserved								
Default :0	GPIO124	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	PVT_CS#	O			All PGS options	NA			
2	Reserved								
3	ICT12	I			All PGS options	Low			
4	GPTP_OUT6	O			All PGS options	NA			
5	Reserved								
Default :0	GPIO125	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	PVT_CLK	O			All PGS options	NA			
2	Reserved								
3	GPTP_OUT5	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO126	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	PVT_IO3	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO130	PIO		VTR2	All PGS options	No Gate	Yes	Yes	

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
1	I2C01_SDA	PIO-24			All PGS options	High			When used as an I2C port, this is a 1.8V I2C port and external pull up should be to 1.8V
2	SLV_SPI_IO0	PIO			All PGS options	Low			
3	TOUT1	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO131	PIO		VTR2	All PGS options	No Gate	Yes	Yes	
1	I2C01_SCL	PIO-24			All PGS options	High			When used as an I2C port, this is a 1.8V I2C port and external pull up should be to 1.8V
2	SLV_SPI_CS#	I			All PGS options	High			
3	TOUT0	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO145	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	I2C09_SDA	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
4	JM_TDI	O			All PGS options	NA			
5	Reserved								
Default :0	GPIO146	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	I2C09_SCL	PIO			All PGS options	High			
2	ITM	O			All PGS options	Low			
3	Reserved								
4	JM_TDO	I			All PGS options	Low			
5	Reserved								
Default :0	GPIO147	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	I2C15_SDA	PIO			All PGS options	High			
2	Reserved								

© 2021 Microchip Technology Inc.

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
3	Reserved								
4	JM_TCLK	O			All PGS options	NA			
5	Reserved								
Default :0	GPIO150	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	I2C15_SCL	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
4	JM_TMS	O			All PGS options	NA			
5	Reserved								
Default :0	GPIO152	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	GPTP_OUT3	O			All PGS options	NA			
3	I2C07_SDA_ALT	PIO			All PGS options	High			
4	Reserved								
5	Reserved								
Default :0	GPIO156	PIO		VTR1	All PGS options	No Gate	Yes	Yes	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration
1	LED0	O			All PGS options	NA			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO165	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	32KHZ_IN	I			PGS=00 (only)	Low			
2	Reserved								
3	CTOUT0	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO170	PIO	PU	VTR1	All PGS options	No Gate	Yes	Yes	

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
1	UART1_TX	O			All PGS options	NA			
2	TFDP_CLK	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO171	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	UART1_RX	I			All PGS options	Low			
2	TFDP_DATA	O			All PGS options	NA			
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO175	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	CMP_VOUT1	O			All PGS options	NA			
2	Reserved								
3	PWM8_ALT	O			All PGS options	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO200	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC00	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO201	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC01	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO202	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC02	I_AN			PGS=00 (only)	Low			
2	Reserved								

© 2021 Microchip Technology Inc.

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO203	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC03	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO204	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC04	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO205	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC05	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO206	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC06	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO207	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	ADC07	I_AN			PGS=00 (only)	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO221	PIO		VTR1	All PGS options	No Gate	Yes	No	

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
1	32KHz_OUT	O			All PGS options	NA			
2	GPTP_IN3	I			All PGS options	Low			
3	CMP_VIN1	I_AN			PGS=00 (only)	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO224	PIO-24	24mA	VTR2	All PGS options	No Gate	Yes	Yes	Drive strength can be configured by Pin Control register2 to 4,8,16 or 24mA
1	GPTP_IN0	I			All PGS options	Low			
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
Default :0	GPIO226	PIO		VTR1	All PGS options	No Gate	Yes	No	
1	Reserved								
2	Reserved								
3	CMP_VREF0	I_AN			PGS=00 (only)	NA			
4	Reserved								
5	Reserved								
Default :0	GPIO241	PIO		VTR1	All PGS options	No Gate	Yes	Yes	
1	Reserved								
2	Reserved								
3	CMP_VOUT0	O			PGS=00 (only)	NA			
4	PWM0_ALT	O			All PGS options	NA			
5	Reserved								
Default :1	JTAG_RST#	I		VTR1	NA	NA	Yes	Yes	
0	Reserved								
2	Reserved								
3	Reserved								
4	Reserved								
5	Reserved								
NA	nRESET_IN	I		VTR1	NA	No Gate	Yes	Yes	
1	Reserved								
2	Reserved								

© 2021 Microchip Technology Inc.

TABLE 2-2: EEC1727 68 WFBGA PIN MUX TABLE (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OVP	BDP	Notes
3	Reserved								
4	Reserved								
5	Reserved								

2.4.11 CONFIGURABLE SIGNAL ROUTING

Host interface signals, nEC_SCI and nSMI are routed on 2 pins, one that's powered by VTR1 and other VTR3. This is to accommodate 3.3/1.8V signaling on these based on the Host. The signal routing is determined by the alternate function multiplexer programmed in the pin's GPIO Pin Control Register.Software should not enable signals on more than one pin.

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.

TABLE 2-3: GPIO ALTERNATE FUNCTIONS

Function	GPIO <Signal>	Alternate GPIO <Signal>_ALT	Second Alternate GPIO <Signal>_ALT2
32kHz_OUT	GPIO221	GPIO022	
I2C07_SCL	GPIO013	GPIO024	
I2C07_SDA	GPIO012	GPIO152	
PWM8	GPIO035	GPIO175	
PWM0	GPIO053	GPIO241	

2.4.12 SIGNAL DESCRIPTION BY INTERFACE

TABLE 2-4: PIN DESCRIPTION TABLE

SIG_NAME	Description	Notes
ADC		
ADCxx	ADC channel x	Note 1
VREF_ADC	ADC Reference Voltage	
VREF2_ADC	Alternate Vref for ADC	
MailBox		
nSMI	SMI output	
PWM LED		
LEDx	LED (Blinking/Breathing PWM) PWM Output x	Note 1
Debug		
TFDP_CLK	Trace FIFO debug port - clock	
TFDP_DATA	Trace FIFO debug port - data	
JTAG_RST#	JTAG test active low reset	Note 8
JTAG_TDI	JTAG test data in	Note 10,Note 11,Note 14
JTAG_TDO	JTAG test data out	Note 10,Note 11,Note 14, Note 17, Note 19

TABLE 2-4: PIN DESCRIPTION TABLE (CONTINUED)

SIG_NAME	Description	Notes
JTAG_CLK	JTAG test clk; SWDCLK	Note 10,Note 11,Note 14, Note 18, Note 19
JTAG_TMS	JTAG test mode select; SWDIO	Note 10,Note 11,Note 14, Note 17, Note 18, Note 19
JM_TDI	Muxed on JTAG_TDI pin JTAG Master TDI	Note 14
JM_TDO	Muxed on JTAG_TDO pin JTAG Master TDO	Note 14
JM_CLK	Muxed on JTAG_TCK pin JTAG Master TCK	Note 14
JM_TMS	Muxed on JTAG_TMS pin JTAG Master TMS	Note 14
TRACECLK	ARM Embedded Trace Macro Clock	
TRACEDATAx	ARM Embedded Trace Macro Data x	Note 1
Slave SPI		
SLV_SPI_CS#	Slave SPI Chip Select	Note 8
SLV_SPI_SCLK	Slave SPI Clock	Note 15
SLV_SPI_IOx	Slave SPI Data x	Note 1
SLV_SPI_MSTR_INT	Slave SPI interrupt to Master	
SMBus/I2C Controller		
I2Cxx_SDA	I2C/SMBus Controller Port x Data	Note 1,Note 3
I2Cxx_SCL	I2C/SMBus Controller Port x Clock	
Analog Comparator		
CMP_VIN0	Comparator 0 Positive Input	
CMP_VIN1	Comparator 1 Positive Input	
CMP_VOUT0	Comparator 0 Output	
CMP_VOUT1	Comparator 1 Output	
CMP_VREF0	Comparator 0 Negative Input	
CMP_VREF1	Comparator 1 Negative Input	
GPIO		
GPIOx	General Purpose Input Output Pins	Note 1
GPTP_INx	General purpose pass through port inputx	Note 1,Note 12
GPTP_OUTx	General purpose pass through port outputx	Note 1,Note 12
PCR		
32KHZ_OUT	32.768 KHz Digital Output	
nRESET_IN	External System Reset Input	
QMSPI		
PVT_CS#	Private SPI Chip Select; SPI_CS# of QMSPI Controller	Note 8
PVT_IOx	Private SPI Data x; SPI_IOx of QMSPI Controller	Note 1
PVT_CLK	Private SPI Clock; SPI_CLK of QMSPI Controller	Note 15
FAN TACH		
ICT0_TACH0	Fan Tachometer Input 0	
ICT1_TACH1	Fan Tachometer Input 1	
GPWMx	PWM Output from RPM-based Fan Speed Control Algorithm	Note 1
PWM		
PWMx	Pulse Width Modulator Output x	Note 1

© 2021 Microchip Technology Inc.

TABLE 2-4: PIN DESCRIPTION TABLE (CONTINUED)

SIG_NAME	Description	Notes
ICT		
ICTx	Input capture timer input x	Note 1
CTOUTx	Compare timer x toggle output	Note 1
16-Bit Counter/Timer Interface		
TOUTx	16-Bit Counter/Timer Outputx	Note 1
UART		
UARTx_RX	UART Receive Data (RXD)	Note 1
UARTx_TX	UART Transmit Data (TXD)	Note 1
Power		
VTR_ANALOG	Analog supply	
VSS_ANALOG	Analog Supply associated ground	
VSS_ADC	Analog ADC supply associated ground	
VR_CAP	Internal Voltage Regulator Capacitor	Note 2
VSSx	VTRx associated ground	Note 1
VTR1	VTR Suspend Power Supply	
VTR2	Peripheral Power Supply	
VTR_PLL	PLL power supply	
VTR_REG	Regulator power supply	
Notes for the Pin description table		

TABLE 2-4: PIN DESCRIPTION TABLE (CONTINUED)

SIG_NAME	Description	Notes

Note 1: x' is the number of the signals in the chip. Please refer Pin List in the Pin configuration chapter to know the number of signals in each interface available

2: An external cap must be connected as close to the 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.

3: 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, Table 20-5 Bus Capacitance/Pull-Up Resistor Relationship.

4: 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

5: The XTAL1 pin should be left floating when using the XTAL2 pin for the single ended clock input.

6: This signal is a test signal used to detect when the internal 48MHz clock is toggling or stopped in heavy and deepest sleep modes.

7: The nEC_SCI pin can be controlled by hardware and EC firmware. The nEC_SCI pin can drive either the ACPI Run-time GPE Chipset input or the Wake GPE Chipset input. Depending how the nEC_SCI pin is used, other ACPI-related SCI functions may be best supplied by other general purpose outputs that can be configured as open-drain drivers.

8: <Signal> with '#' as suffix will be shown as <Signal>_n in MPLab Tools

9: PS/2 ports ending with signal functions ending with "A" or "B" are muxed to a single controller. Only one set of clock and data are intended to used at a time (either "A" or "B" not both. The unused port segment should have its associated pin control register's, Mux Control Field programmed away from the PS2 controller.

10: 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. 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

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.

12: The GPTP_OUT always drives at the level of the output buffer regardless of the voltage at the GPTP_IN pin. If the GPTP_IN pin is 1.8V the output essentially level-shifts the voltage up to 3.3V, as GPTP_OUT pins are powered by VTR1 (3.3V)

13: These VCI pins may be used as GPIOs as well. 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

14: External Pullup is required on the JTAG pins when used for debug operation

15: The maximum clock frequency of this interface is 48MHz

16: The maximum clock frequency of this interface is 96MHz for single SPI. If the SPI flash is shared by more than one master, the maximum clock frequency of this interface is 48MHz.

17: The 2-Pin JTAG mode uses JTAG_TMS and JTAG_TDO pins

18: Serial Wire Debug (SWD) mode uses JTAG_CLK as SWDCLK and JTAG_TMS as SWDIO.

19: Serial Wire Viewer (SWV) mode uses JTAG_CLK, JTAG_TMS and JTAG_TDO pins

© 2021 Microchip Technology Inc.

2.4.13 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

TABLE 2-5: STRAP PINS

Pin Name	Strap Name	Strap Define and Value	I/O Power Rail
GPIO170	JTAG_STRAP	1=Use the JAG TAP Controller for Boundary Scan 0=The JTAG TAP Controller is used for debug (normal operation)	VTR1
GPIO165	CR_STRAP	Crisis Recovery Strap 1=Normal Boot Source 0=Use the Private SPI pins to boot from Crisis Recovery flash Note: This pin requires an external pull-up for normal operation.	VTR1
GPIO207	CMP_STRAP	CMP_STRAP is the Comparator 0 Strap pin. This strap must be enabled in OTP.Note 1 1=Comparator 0 Enabled. 0=Hardware Default (GPIO input)	VTR1
GPIO126	UART_BSTRAP	Crisis Recovery over UART 1=Normal Operation 0=Use UART interface for Crisis recovery Note 2	VTR1

1: The comparator strap option is an optional feature that may be enabled in OTP to enable the Boot ROM to configure and lock the Comparator 0 pins. If the feature is enabled in OTP, and external pull-up/pull-down is required to determine the default comparator behavior. If the strap option is not enabled in OTP, the CMP_STRAP is not supported and no external pull-up or pull-down required. Application firmware may enable the comparator if supported by the specific package.

2: OTP byte 115 bit [3] allows selection of UART0/UART1 for Crisis Recovery, if UART_BSTRAP is enabled and sampled as 0x0.

2.5 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	VTR Applied	RESET_SYS De-asserted	VCC_PWRGD Asserted	VCC_PWRGD De-asserted	RESET_SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
GPIO170	High	In	In	In	Z	glitch	un-powered	
GPIOXXX	Z	Z	Z	Z	Z	glitch	un-powered	Note D
nRESET_IN	Low	In	In	In	Z	glitch	un-powered	

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.

TABLE 2-7: PIN DEFAULT STATE THROUGH POWER TRANSITIONS

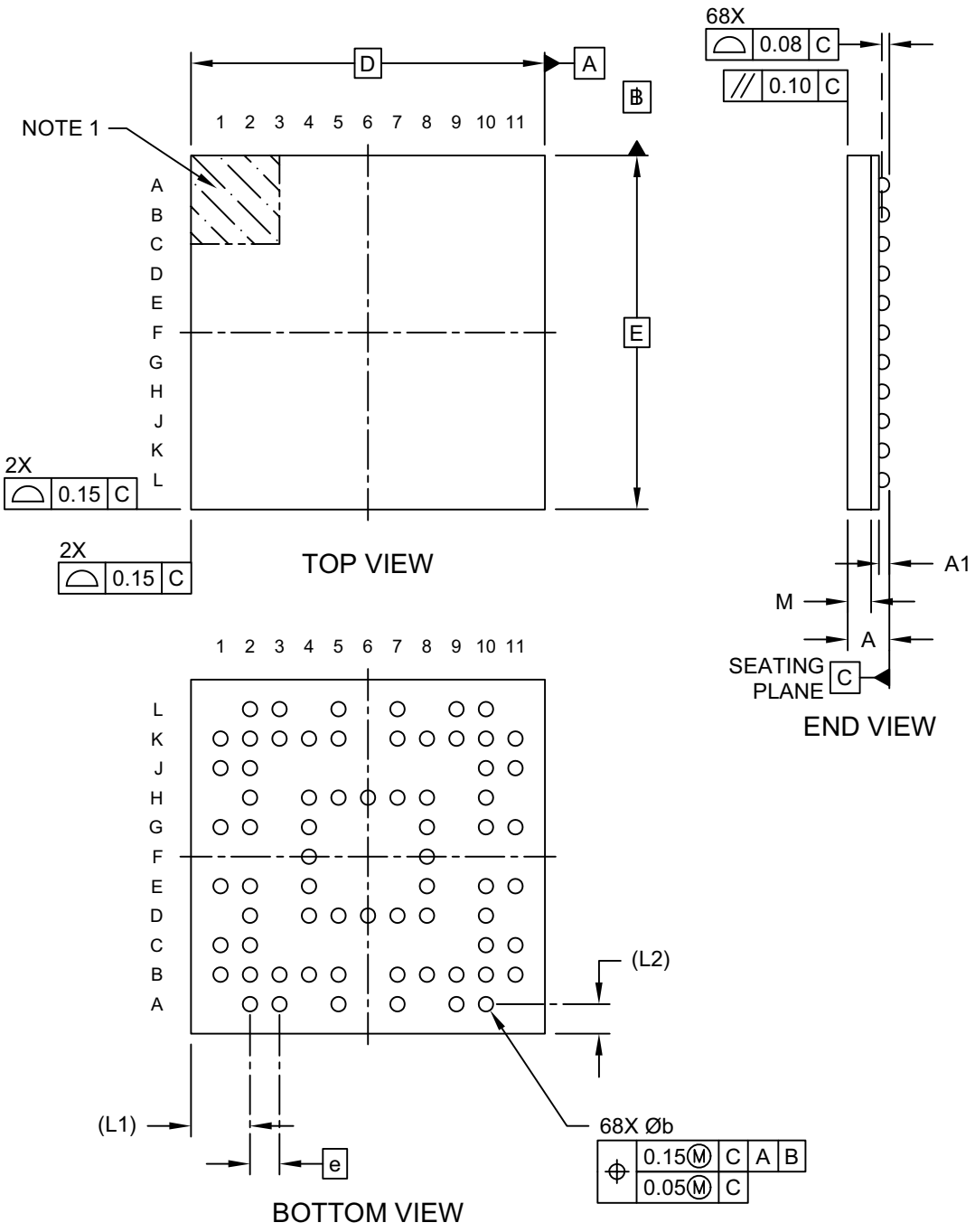
Signal	VBAT Applied	VBAT Stable	VTR Applied	RESET_SYS De-asserted	VCC_PWRGD Asserted	VCC_PWRGD De-asserted	RESET_SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
Legend **(P)** = I/O state is driven by protocol while power is applied. **Z** = Tristate **In** = Input **OD** = Open Drain Output Undriven (1) or driven (0)			Notes Note F: Pin is programmable by the EC and retains its value through a VTR power cycle							

© 2021 Microchip Technology Inc.

2.6 Package Information

68-Ball Very, Very Thin Fine Pitch Ball Grid Array (2GW) - 6x6x0.8 mm Body [WFBGA]

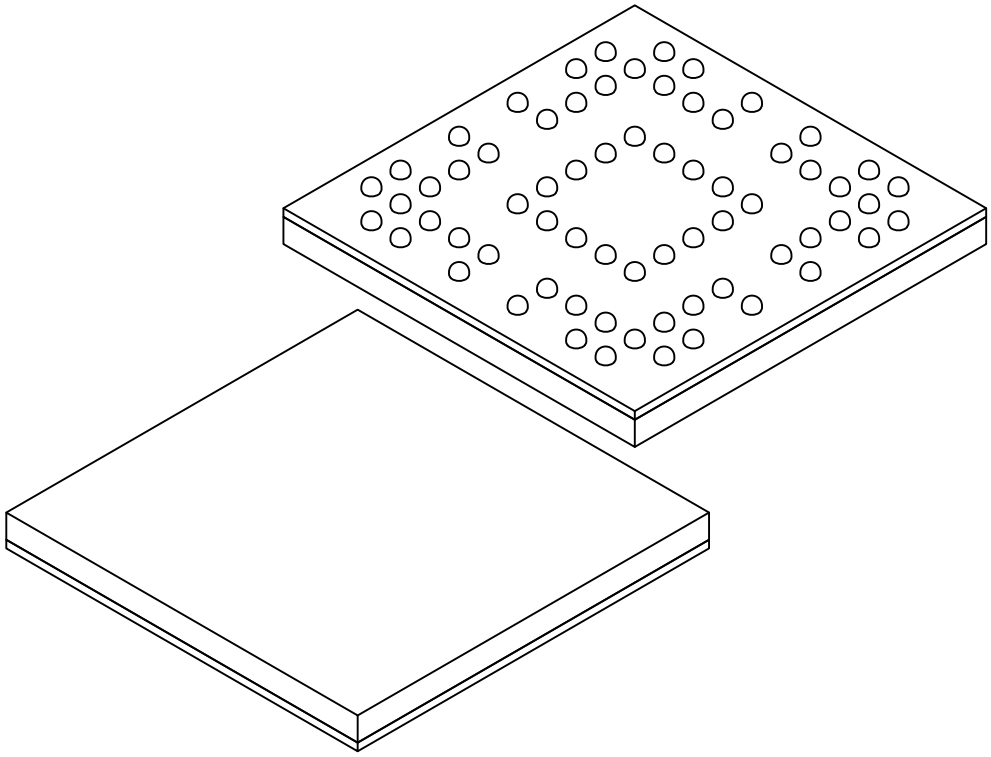
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-542 Rev A Sheet 1 of 2

68-Ball Very, Very Thin Fine Pitch Ball Grid Array (2GW) - 6x6x0.8 mm Body [WFBGA]

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		68	
Pitch	e		0.50 BSC	
Overall Height	A	–	–	0.80
Ball Height	A1	0.13	0.18	0.23
Mold Thickness	M	0.35	0.40	0.45
Overall Length	D		6.00 BSC	
Overall Width	E		6.00 BSC	
Ball Diameter	b	0.20	0.25	0.30
Edge to Ball Center	L1		1.00 REF	
Edge to Ball Center	L2		0.50 REF	

Notes:

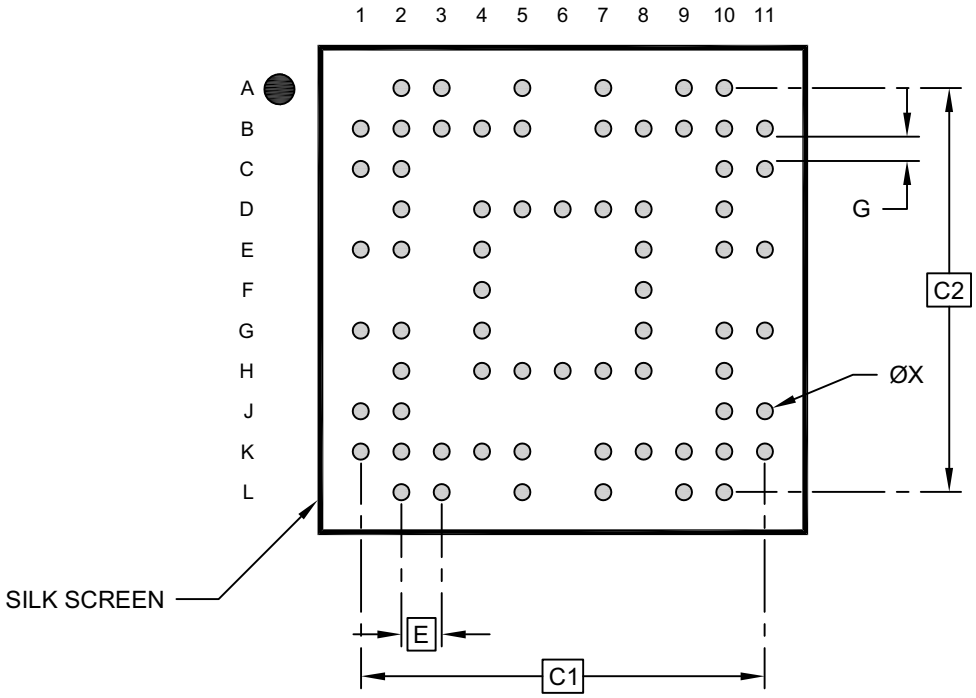
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-542 Rev A Sheet 2 of 2

© 2021 Microchip Technology Inc.

68-Ball Very, Very Thin Fine Pitch Ball Grid Array (2GW) - 6x6x0.8 mm Body [WFBGA]

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	0.50 BSC		
Contact Pad Spacing	C1	5.00 BSC		
Contact Pad Spacing	C2	5.00 BSC		
Contact Pad Diameter (X68)	X			0.20
Contact Pad to Contact Pad	G	0.30		

Notes:

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

Microchip Technology Drawing C04-2542 Rev A

© 2021 Microchip Technology Inc.

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.

© 2021 Microchip Technology Inc.

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
16-bit Basic Timer	2		4000_0C40h
16-bit Basic Timer	3		4000_0C60h
32-bit Basic Timer	0		4000_0C80h
32-bit Basic Timer	1		4000_0CA0h
16-bit Counter Timer	0		4000_0D00h
16-bit Counter Timer	1		4000_0D20h
16-bit Counter Timer	2		4000_0D40h
16-bit Counter Timer	3		4000_0D60h
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
Quad Master SPI			4007_0000h
16-bit PWM	0		4000_5800h
16-bit PWM	1		4000_5810h
16-bit PWM	2		4000_5820h
16-bit PWM	3		4000_5830h
16-bit PWM	4		4000_5840h
16-bit PWM	5		4000_5850h
16-bit PWM	6		4000_5860h
16-bit PWM	7		4000_5870h
16-bit PWM	8		4000_5880h
16-bit PWM	9		4000_5890h
16-bit PWM	10		4000_58A0h
16-bit PWM	11		4000_58B0h
16-bit Tach	0		4000_6000h
16-bit Tach	1		4000_6010h
16-bit Tach	2		4000_6020h
16-bit Tach	3		4000_6030h
RTOS Timer			4000_7400h
ADC			4000_7C00h

© 2021 Microchip Technology Inc.

TABLE 3-1: BASE ADDRESS

Feature	Instance	Logical Device Number	Base Address
Trace FIFO			4000_8C00h
Hibernation Timer	0		4000_9800h
Hibernation Timer	1		4000_9820h
RPM2PWM	0		4000_A000h
RPM2PWM	1		4000_A080h
VBAT Register Bank			4000_A400h
VBAT Powered RAM			4000_A800h
Week Timer			4000_AC80h
Blinking-Breathing LED	0		4000_B800h
Blinking-Breathing LED	1		4000_B900h
Blinking-Breathing LED	2		4000_BA00h
Blinking-Breathing LED	3		4000_BB00h
Interrupt Aggregator			4000_E000h
EC Subsystem Registers			4000_FC00h
JTAG			4008_0000h
Power, Clocks and Resets (PCR)			4008_0100h
GPIOs			4008_1000h
Port 92-Legacy		8h	400F_2000h
UART	0	9h	400F_2400h
UART	1	Ah	400F_2800h
Real Time Clock		14h	400F_5000h
Global Configuration		3Fh	400F_FF00h
SPI Slave			4000_7000h
Cache Controller			4000_5400h

Note: The 32-Bit BIOS Debug Port contains two Logical Address register sets. The basic functionality supports 4 contiguous I/O bytes in the first ("Base") Logical Device register set (Logical Device 20h). One of these bytes can also be aliased to a non-contiguous I/O location, for legacy 16-bit Port 80 display handling, by using the separate "Alias" Logical Device register set (21h).

© 2021 Microchip Technology Inc.

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
ISPI		2	NA	Clock Required 0	Reset Enable 0
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
PMC/CPP reg Bank		5	Sleep Enable 1	Clock Required 1	NA
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
Tach	2	12	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	3	13	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	1	20	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	4	23	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
PWM	8	27	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
IMAP	0	0	NA	Clock Required 2	Reset Enable 2
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
ACPI EC	0	13	NA	NA	Reset Enable 2
ACPI EC	1	14	NA	NA	Reset Enable 2
ACPI PM1		15	NA	NA	Reset Enable 2
8042 Emulation		16	NA	Clock Required 2	Reset Enable 2
Mailbox		17	NA	NA	Reset Enable 2
RTC		18	NA	Clock Required 2	NA
ACPI EC	2	21	NA	Clock Required 2	Reset Enable 2
ACPI EC	3	22	NA	Clock Required 2	Reset Enable 2
ACPI EC	4	23	NA	Clock Required 2	Reset Enable 2
Port 80	0	25	NA	NA	Reset Enable 2
SAF_BRIDGE		27	NA	Clock Required 2	NA

TABLE 3-2: SLEEP ALLOCATION

Block	Instance	Bit Position	Sleep Enable Register	Clock Required Register	Reset Enable Register
ADC		3	Sleep Enable 3	Clock Required 3	Reset Enable 3
Hibernation Timer	0	10	Sleep Enable 3	Clock Required 3	Reset Enable 3
RPM2PWM	0	12	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
LED	2	18	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	4	20	Sleep Enable 3	Clock Required 3	Reset Enable 3
Basic Timer 16	2	21	Sleep Enable 3	Clock Required 3	Reset Enable 3
Basic Timer 16	3	22	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
LED	3	25	Sleep Enable 3	Clock Required 3	Reset Enable 3
Crypto		26	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
PWM	9	31	Sleep Enable 3	Clock Required 3	Reset Enable 3
PWM	10	0	Sleep Enable 4	Clock Required 4	Reset Enable 4
PWM	11	1	Sleep Enable 4	Clock Required 4	Reset Enable 4
16-bit Counter Timer	0	2	Sleep Enable 4	Clock Required 4	Reset Enable 4
16-bit Counter Timer	1	3	Sleep Enable 4	Clock Required 4	Reset Enable 4
16-bit Counter Timer	2	4	Sleep Enable 4	Clock Required 4	Reset Enable 4
16-bit Counter Timer	3	5	Sleep Enable 4	Clock Required 4	Reset Enable 4
RTOS Timer		6	NA	Clock Required 4	Reset Enable 4
RPM2PWM	1	7	Sleep Enable 4	Clock Required 4	Reset Enable 4
Quad SPI Master		8	Sleep Enable 4	Clock Required 4	Reset Enable 4
SLV_SPI	0	16	NA	Clock Required 4	Reset Enable 4

© 2021 Microchip Technology Inc.

3.4 Interrupt Aggregator Bit Assignments

TABLE 3-3: GPIO 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	GPIO141	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO142	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO151	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO152	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO153	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO154	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO155	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO156	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO157	GPIO Event	Yes	GPIO Interrupt Event		
	16	GPIO160	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO161	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO162	GPIO Event	Yes	GPIO Interrupt Event		
	19-20	Reserved					
	21	GPIO165	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO166	GPIO Event	Yes	GPIO Interrupt Event		
	23	Reserved					
	24	GPIO170	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO171	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO172	GPIO Event	Yes	GPIO Interrupt Event		
	29	GPIO175	GPIO Event	Yes	GPIO Interrupt Event		
	30	Reserved	-	-			
	31	Reserved	-	-			
GIRQ9	0	GPIO100	GPIO Event	Yes	GPIO Interrupt Event	1	N/A
	1	GPIO101	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO102	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO103	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO110	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO111	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO112	GPIO Event	Yes	GPIO Interrupt Event		

© 2021 Microchip Technology Inc.

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	11	GPIO113	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO114	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO115	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO117	GPIO Event	Yes	GPIO Interrupt Event		
	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		
	20	GPIO124	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO133	GPIO Event	Yes	GPIO Interrupt Event		
	28	GPIO134	GPIO Event	Yes	GPIO Interrupt Event		
	29	GPIO135	GPIO Event	Yes	GPIO Interrupt Event		
	30	Reserved	-	-			
	31	Reserved	-	-			
GIRQ10	0	GPIO040	GPIO Event	Yes	GPIO Interrupt Event	2	N/A
	1	GPIO041	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO042	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO043	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO044	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO052	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO053	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO054	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO060	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO061	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO063	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO064	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO065	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO066	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO067	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO070	GPIO Event	Yes	GPIO Interrupt Event		

© 2021 Microchip Technology Inc.

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	25	GPIO071	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO072	GPIO Event	Yes	GPIO Interrupt Event		
	27	GPIO073	GPIO Event	Yes	GPIO Interrupt Event		
	28	GPIO074	GPIO Event	Yes	GPIO Interrupt Event		
	29	GPIO075	GPIO Event	Yes	GPIO Interrupt Event		
	30	GPIO076	GPIO Event	Yes	GPIO Interrupt Event		
	31	Reserved	-	-			
GIRQ11	0	GPIO000	GPIO Event	Yes	GPIO Interrupt Event	3	N/A
	1	GPIO001	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO005	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO006	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO007	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO010	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO011	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO012	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO013	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO014	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO015	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO016	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO017	GPIO Event	Yes	GPIO Interrupt Event		
	16	GPIO020	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO021	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO022	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO023	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO024	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO025	GPIO Event	Yes	GPIO Interrupt Event		
	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	GPIO031	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO032	GPIO Event	Yes	GPIO Interrupt Event		
	27	GPIO033	GPIO Event	Yes	GPIO Interrupt Event		
	28	GPIO034	GPIO Event	Yes	GPIO Interrupt Event		
	29	GPIO035	GPIO Event	Yes	GPIO Interrupt Event		
	30	GPIO036	GPIO Event	Yes	GPIO Interrupt Event		
	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		

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	5	GPIO205	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO206	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO207	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO210	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO211	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO212	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO213	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO214	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO215	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO216	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO217	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO221	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO222	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO223	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO224	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO225	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO226	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO227	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO230	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO231	GPIO Event	Yes	GPIO Interrupt Event		
	27	GPIO233	GPIO Event	Yes	GPIO Interrupt Event		
	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
	5-31	Reserved	-	-			
GIRQ14	0	DMA Controller	DMA0	No	DMA Controller - Channel 0 Interrupt Event	6	24
	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

© 2021 Microchip Technology Inc.

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	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	DMA Controller	DMA12	No	DMA Controller - Channel 12 Interrupt Event		36
	13	DMA Controller	DMA13	No	DMA Controller - Channel 13 Interrupt Event		37
	14	DMA Controller	DMA14	No	DMA Controller - Channel 11 Interrupt Event		38
	15	DMA Controller	DMA15	No	DMA Controller - Channel 12 Interrupt Event		39
	16-31	Reserved					
GIRQ15	0	UART 0	UART	No	UART Interrupt Event	7	40
	1	UART 1	UART	No	UART Interrupt Event		41
	2	EMI 0	Host-to-EC	No	Embedded Memory Interface 0 - Host-to-EC Interrupt		42
	3	EMI 1	Host-to-EC	No	Embedded Memory Interface 1 - Host-to-EC Interrupt		43
	4	EMI 2	Host-to-EC	No	Embedded Memory Interface 1 - Host-to-EC Interrupt		44
	5-19	Reserved					
	5	ACPI EC Interface 0	IBF	No	ACPI EC Interface 0 - Input Buffer Full Event		45
	6	ACPI EC Interface 0	OBE	No	ACPI EC Interface 0 - Output Buffer Empty Event, asserted when OBE flag goes to 1		46
	7	ACPI EC Interface 1	IBF	No	ACPI EC Interface 1 - Input Buffer Full Event		47
	8	ACPI EC Interface 1	OBE	No	ACPI EC Interface 1 - Output Buffer Empty Event, asserted when OBE flag goes to 1		48
	9	ACPI EC Interface 2	IBF	No	ACPI EC Interface 2 - Input Buffer Full Event		49
	10	ACPI EC Interface 2	OBE	No	ACPI EC Interface 2 - Output Buffer Empty Event, asserted when OBE flag goes to 1		50
	11	ACPI EC Interface 3	IBF	No	ACPI EC Interface 3 - Input Buffer Full Event		51

© 2021 Microchip Technology Inc.

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	12	ACPI EC Interface 3	OBE	No	ACPI EC Interface 3 - Output Buffer Empty Event, asserted when OBE flag goes to 1		52
	13	ACPI EC Interface 4	IBF	No	ACPI EC Interface 4 - Input Buffer Full Event		53
	14	ACPI EC Interface 4	OBE	No	ACPI EC Interface 4 - Output Buffer Empty Event, asserted when OBE flag goes to 1		54
	15	ACPI_PM1	PM1_CTL	No	ACPI_PM1 Interface - PM1_CTL2 Interrupt Event		55
	16	ACPI_PM1	PM1_EN	No	ACPI_PM1 Interface - PM1_EN2 Interrupt Event		56
	17	ACPI_PM1	PM1_STS	No	ACPI_PM1 Interface - PM1_STS2 Interrupt Event		57
	20	Mailbox	MBX	No	Mailbox Interface - Host-to-EC Interrupt Event		60
	22	32 bit Port 80 Debug 0	BDP_INT	No	Port 80h BIOS Debug Port Event		62
	21-24	Reserved		-	-		
	25-31	Reserved		-	-		
GIRQ16	0	Public Key Engine	PKE _INT	No	PKE Interrupt	8	65
	1	Reserved					
	2	Random Number Generator	TRNG_INT	No	TRNG completed processing		67
	3	AES-HASH	AES_HASH _INT	No	Interrupt from AES or SHA block		68
	4-31	Reserved					
GIRQ17	0	Reserved				9	
	1	TACH 0	TACH	No	Tachometer 0 Interrupt Event		71
	2	TACH 1	TACH	No	Tachometer 1 Interrupt Event		72
	3	TACH 2	TACH	No	Tachometer 2 Interrupt Event		73
	4	TACH3	TACH	No	Tachometer 3 Interrupt Event		159
	5-7	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	Breathing LED 2	PWM_WDT	No	Blinking LED 2 Watchdog Event		85
	16	Breathing LED 3	PWM_WDT	No	Blinking LED 3 Watchdog Event		86
	20	RPM2PWM 0	FAN_FAIL	No	Failure to achieve target RPM		74
	21	RPM2PWM 0	FAN_STALL	No	Fan stall condition		75
	22	RPM2PWM 1	FAN_FAIL	No	Failure to achieve target RPM		76
	23	RPM2PWM 1	FAN_STALL	No	Fan stall condition		77

© 2021 Microchip Technology Inc.

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	24-31	Reserved	-	-			
GIRQ18	0	Slave SPI	SPI_EC_INTERRUPT	No	Slave SPI Interrupt	10	90
	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	0-31	Reserved					
GIRQ20	0-8	Test	Test	-	-	12	N/A
	0	STAP	STAP_OBF	No	Debug Output Buffer FIFO is Empty		N/A
	1	STAP	STAP_IBF	No	Debug Input Buffer FIFO is Full		
	2	STAP	STAP_WAKE	Yes	STAP Initiated Wake Event		
	3	OTP	READY_INTR	No	OTP ready interrupt		173
	4-7	Reserved					
	8	ISPI	ISPI_ERROR	No	ISPI Error		
	9	32KHz Clock Monitor	CLK_32KHZ_MONITOR	No	32KHz Clock Counter Monitor		174
	10-31	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

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	6	Week Alarm	SUB_SEC-OND	Yes	Week Alarm - Sub-second Interrupt		117
	7	Week Alarm	SYS-PWR_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	VBAT-Powered Control Interface	VCI_OVRD_IN	Yes	VCI_OVRD_IN active high input pin interrupt		121
	11	VBAT-Powered Control Interface	VCI_IN0	Yes	VCI_IN0 Active-low Input Pin Interrupt		122
	12	VBAT-Powered Control Interface	VCI_IN1	Yes	VCI_IN1 Active-low Input Pin Interrupt		123
	16-17	Reserved					
	10-24	Reserved					
	25	Reserved					
	24-31	Reserved					
GIRQ22	0	Slave SPI	SPI_ASYN-C_WAKE	Yes	Wake-Only Event		N/A
	1	SMB-I2C Con-troller0	SMB-I2C _WAKE_ON LY	Yes	Wake-Only Event (No Interrupt Gener-ated) - SMB-I2C.0 START Detected		
	2	SMB-I2C Con-troller1	SMB-I2C _WAKE_ON LY	Yes	Wake-Only Event (No Interrupt Gener-ated) - SMB-I2C.1 START Detected		
	3	SMB-I2C Con-troller2	SMB-I2C _WAKE_ON LY	Yes	Wake-Only Event (No Interrupt Gener-ated) - SMB-I2C.2 START Detected		
	4	SMB-I2C Con-troller3	SMB-I2C _WAKE_ON LY	Yes	Wake-Only Event (No Interrupt Gener-ated) - SMB-I2C.3 START Detected		
	5	SMB-I2C Con-troller4	SMB-I2C _WAKE_ON LY	Yes	Wake-Only Event (No Interrupt Gener-ated) - SMB-I2C.4 START Detected		
	6-8	Reserved	-	-			
	9-14	Reserved		-			
	15	STAP	STAP_WAK E	Yes	STAP Initiated Wake Event		
	16-31	Reserved		-			
GIRQ23	0	16-Bit Basic Timer 0	Timer_Event	No	Basic Timer Event	14	136
	1	16-Bit Basic Timer 1	Timer_Event	No	Basic Timer Event		137
	2	16-Bit Basic Timer 2	Timer_Event	No	Basic Timer Event		138
	3	16-Bit Basic Timer 3	Timer_Event	No	Basic Timer Event		139
	4	32-Bit Basic Timer 0	Timer_Event	No	Basic Timer Event		140

© 2021 Microchip Technology Inc.

TABLE 3-3: GPIO MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	5	32-Bit Basic Timer 1	Timer_Event	No	Basic Timer Event		141
	6	Counter/Timer 0	Timer_Event	No	16-bit Timer/Counter Event		142
	7	Counter/Timer 1	Timer_Event	No	16-bit Timer/Counter Event		143
	8	Counter/Timer 2	Timer_Event	No	16-bit Timer/Counter Event		144
	9	Counter/Timer 3	Timer_Event	No	16-bit Timer/Counter Event		145
	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
	18-31	Reserved					
GIRQ24	0-31	Reserved					
GIRQ25	0-31	Reserved					
Direct NVIC Interrupts							
		ACPI EC Interface 0	ACPIEC_CMN_INT0				175
		ACPI EC Interface 1	ACPI EC_CMN_INT1				176
		ACPI EC Interface 2	ACPIEC_CMN_INT2				177
		ACPI EC Interface 3	ACPIEC_CMN_INT3				178
		ACPI EC Interface 4	ACPI EC_CMN_INT4				179
		ACPI_PM1	ACPIPM1_CMN_INT				180

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
GPIO170	0x00000041	JTAG_STRAP BS (input, pull up)

© 2021 Microchip Technology Inc.

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 Int 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
16-bit Basic Timer	2	Timer Count Register		40000C40
16-bit Basic Timer	2	Timer Preload Register		40000C44
16-bit Basic Timer	2	Timer Status Register		40000C48
16-bit Basic Timer	2	Timer Int Enable Register		40000C4C
16-bit Basic Timer	2	Timer Control Register		40000C50
16-bit Basic Timer	3	Timer Count Register		40000C60
16-bit Basic Timer	3	Timer Preload Register		40000C64
16-bit Basic Timer	3	Timer Status Register		40000C68
16-bit Basic Timer	3	Timer Int Enable Register		40000C6C
16-bit Basic Timer	3	Timer Control Register		40000C70
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
16-bit Counter Timer	0	Timer x Control Register		40000D00
16-bit Counter Timer	0	Timer x Clock and Event Control Register		40000D04
16-bit Counter Timer	0	Timer x Reload Register		40000D08
16-bit Counter Timer	0	Timer x Count Register		40000D0C

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
16-bit Counter Timer	1	Timer x Control Register		40000D20
16-bit Counter Timer	1	Timer x Clock and Event Control Register		40000D24
16-bit Counter Timer	1	Timer x Reload Register		40000D28
16-bit Counter Timer	1	Timer x Count Register		40000D2C
16-bit Counter Timer	2	Timer x Control Register		40000D40
16-bit Counter Timer	2	Timer x Clock and Event Control Register		40000D44
16-bit Counter Timer	2	Timer x Reload Register		40000D48
16-bit Counter Timer	2	Timer x Count Register		40000D4C
16-bit Counter Timer	3	Timer x Control Register		40000D60
16-bit Counter Timer	3	Timer x Clock and Event Control Register		40000D64
16-bit Counter Timer	3	Timer x Reload Register		40000D68
16-bit Counter Timer	3	Timer x Count Register		40000D6C
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
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
DMA Channel	12	DMA Channel N Activate Register		40002740
DMA Channel	12	DMA Channel N Memory Start Address Register		40002744
DMA Channel	12	DMA Channel N Memory End Address Register		40002748
DMA Channel	12	DMA Channel N Device Address		4000274C
DMA Channel	12	DMA Channel N Control Register		40002750
DMA Channel	12	DMA Channel N Interrupt Status Register		40002754
DMA Channel	12	DMA Channel N Interrupt Enable Register		40002758
DMA Channel	12	TEST		4000275C
DMA Channel	13	DMA Channel N Activate Register		40002780
DMA Channel	13	DMA Channel N Memory Start Address Register		40002784
DMA Channel	13	DMA Channel N Memory End Address Register		40002788
DMA Channel	13	DMA Channel N Device Address		4000278C
DMA Channel	13	DMA Channel N Control Register		40002790
DMA Channel	13	DMA Channel N Interrupt Status Register		40002794
DMA Channel	13	DMA Channel N Interrupt Enable Register		40002798
DMA Channel	13	TEST		4000279C
DMA Channel	14	DMA Channel N Activate Register		400027C0
DMA Channel	14	DMA Channel N Memory Start Address Register		400027C4
DMA Channel	14	DMA Channel N Memory End Address Register		400027C8
DMA Channel	14	DMA Channel N Device Address		400027CC
DMA Channel	14	DMA Channel N Control Register		400027D0
DMA Channel	14	DMA Channel N Interrupt Status Register		400027D4
DMA Channel	14	DMA Channel N Interrupt Enable Register		400027D8
DMA Channel	14	TEST		400027DC
DMA Channel	15	DMA Channel N Activate Register		40002800
DMA Channel	15	DMA Channel N Memory Start Address Register		40002804
DMA Channel	15	DMA Channel N Memory End Address Register		40002808
DMA Channel	15	DMA Channel N Device Address		4000260C
DMA Channel	15	DMA Channel N Control Register		40002810
DMA Channel	15	DMA Channel N Interrupt Status Register		40002814
DMA Channel	15	DMA Channel N Interrupt Enable Register		40002818
DMA Channel	15	TEST		4000281C
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
I2C-SMB	0	Block ID Register		40004030
I2C-SMB	0	Revision Register		40004034
I2C-SMB	0	Bit-Bang Control Register		40004038
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	TEST		40004070
I2C-SMB	0	TEST		40004074
I2C-SMB	0	TEST		40004078
I2C-SMB	0	I2C Shadow Data		4000407C
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
I2C-SMB	1	TEST		40004470
I2C-SMB	1	TEST		40004474
I2C-SMB	1	TEST		40004478
I2C-SMB	1	I2C Shadow Data Register		4000447C
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	TEST		40004870
I2C-SMB	2	TEST		40004874
I2C-SMB	2	TEST		40004878
I2C-SMB	2	I2C Shadow Data Register		4000487C
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
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	TEST		40004C70
I2C-SMB	3	TEST		40004C74
I2C-SMB	3	TEST		40004C78
I2C-SMB	3	I2C Shadow Data Register		40004C7C
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
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	TEST		40005070
I2C-SMB	4	TEST		40005074
I2C-SMB	4	TEST		40005078
I2C-SMB	4	I2C Shadow Data Register		4000507C
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
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
QMSPI	0	QMSPI Alias Control Register		400700B0

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
QMSPI	0	QMSPI Mode Alternate1 Register		400700C0
QMSPI	0	QMSPI Local DMA RX Enable Register		40070100
QMSPI	0	QMSPI Local DMA TX Enable Register		40070104
QMSPI	0	QMSPI Local DMA RX Control 0 Register		40070110
QMSPI	0	QMSPI Local DMA RX Start Address 0 Register		40070114
QMSPI	0	QMSPI Local DMA RX Length 0 Register		40070118
QMSPI	0	RESERVED		4007011C
QMSPI	0	QMSPI Local DMA RX Control 1 Register		40070120
QMSPI	0	QMSPI Local DMA RX Start Address 1 Register		40070124
QMSPI	0	QMSPI Local DMA RX Length 1 Register		40070128
QMSPI	0	RESERVED		4007012C
QMSPI	0	QMSPI Local DMA RX Control 2 Register		40070130
QMSPI	0	QMSPI Local DMA RX Start Address 2 Register		40070134
QMSPI	0	QMSPI Local DMA RX Length 2 Register		40070138
QMSPI	0	RESERVED		4007013C
QMSPI	0	QMSPI Local DMA TX Control 0 Register		40070140
QMSPI	0	QMSPI Local DMA TX Start Address 0 Register		40070144
QMSPI	0	QMSPI Local DMA TX Length 0 Register		40070148
QMSPI	0	RESERVED		4007014C
QMSPI	0	QMSPI Local DMA TX Control 1 Register		40070150
QMSPI	0	QMSPI Local DMA TX Start Address 1 Register		40070154
QMSPI	0	QMSPI Local DMA TX Length 1 Register		40070158
QMSPI	0	RESERVED		4007015C
QMSPI	0	QMSPI Local DMA TX Control 2 Register		40070160
QMSPI	0	QMSPI Local DMA TX Start Address 2 Register		40070164
QMSPI	0	QMSPI Local DMA TX Length 2 Register		40070168
QMSPI	0	RESERVED		4007016C
Cache Controller	0	Cache Mode		40005400
Cache Controller	0	Cache SPI Bank		4001000C
Cache Controller	0	Cache Tag Validate		40010010
Cache Controller	0	Cache Tag Validate Address		40010014
Cache Controller	0	Cache Status		40010020
Cache Controller	0	Cache Hit Hi		40010040
Cache Controller	0	Cache Hit Lo		40010044
Cache Controller	0	Cache Miss Hi		40010050
Cache Controller	0	Cache Miss Lo		40010054
Cache Controller	0	Cache Fill Hi		40010060
Cache Controller	0	Cache Fill Lo		40010064
Cache Controller	0	RX Buffer 0		400100C0
Cache Controller	0	RX Buffer 1		400100C4
Cache Controller	0	RX Buffer 2		400100C8
Cache Controller	0	RX Buffer 3		400100CC
Cache Controller	0	RX Buffer 4		400100D0
Cache Controller	0	RX Buffer 5		400100D4

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Cache Controller	0	RX Buffer 6		400100D8
Cache Controller	0	RX Buffer 7		400100DC
Cache Controller	0	RX Buffer 8		400100E0
Cache Controller	0	RX Buffer 9		400100E4
Cache Controller	0	RX Buffer 10		400100E8
Cache Controller	0	RX Buffer 11		400100EC
Cache Controller	0	RX Buffer 12		400100F0
Cache Controller	0	RX Buffer 13		400100F4
Cache Controller	0	RX Buffer 14		400100F8
Cache Controller	0	RX Buffer 15		400100FC
Cache Controller	0	Cache Tag Lock		40010800
Cache Controller	0	Cache Tag Valid		40010C00
Cache Controller	0	Cache Tag 0 Address		40011000
Cache Controller	0	Cache Tag 1 Address		40011004
Cache Controller	0	Cache Tag 2 Address		40011008
Cache Controller	0	Cache Tag 3 Address		4001100C
Cache Controller	0	Cache Tag 4 Address		40011010
Cache Controller	0	Cache Tag 5 Address		40011014
Cache Controller	0	Cache Tag 6 Address		40011018
Cache Controller	0	Cache Tag 7 Address		4001101C
Cache Controller	0	Cache Tag 8 Address		40011020
Cache Controller	0	Cache Tag 9 Address		40011024
Cache Controller	0	Cache Tag 10 Address		40011028
Cache Controller	0	Cache Tag 11 Address		4001102C
Cache Controller	0	Cache Tag 12 Address		40011030
Cache Controller	0	Cache Tag 13 Address		40011034
Cache Controller	0	Cache Tag 14 Address		40011038
Cache Controller	0	Cache Tag 15 Address		4001103C
Cache Controller	0	Cache Tag 16 Address		40011040
Cache Controller	0	Cache Tag 17 Address		40011044
Cache Controller	0	Cache Tag 18 Address		40011048
Cache Controller	0	Cache Tag 19 Address		4001104C
Cache Controller	0	Cache Tag 20 Address		40011050
Cache Controller	0	Cache Tag 21 Address		40011054
Cache Controller	0	Cache Tag 22 Address		40011058
Cache Controller	0	Cache Tag 23 Address		4001105C
Cache Controller	0	Cache Tag 24 Address		40011060
Cache Controller	0	Cache Tag 25 Address		40011064
Cache Controller	0	Cache Tag 26 Address		40011068
Cache Controller	0	Cache Tag 27 Address		4001106C
Cache Controller	0	Cache Tag 28 Address		40011070
Cache Controller	0	Cache Tag 29 Address		40011074
Cache Controller	0	Cache Tag 30 Address		40011078
Cache Controller	0	Cache Tag 31 Address		4001107C

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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	1	PWMx Counter ON Time Register		40005810
16-bit PWM	1	PWMx Counter OFF Time Register		40005814
16-bit PWM	1	PWMx Configuration Register		40005818
16-bit PWM	1	TEST		4000581C
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	4	PWMx Counter ON Time Register		40005840
16-bit PWM	4	PWMx Counter OFF Time Register		40005844
16-bit PWM	4	PWMx Configuration Register		40005848
16-bit PWM	4	TEST		4000584C
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 PWM	8	PWMx Counter ON Time Register		40005880
16-bit PWM	8	PWMx Counter OFF Time Register		40005884
16-bit PWM	8	PWMx Configuration Register		40005888
16-bit PWM	8	TEST		4000588C
16-bit PWM	9	PWMx Counter ON Time Register		40005890
16-bit PWM	9	PWMx Counter OFF Time Register		40005894
16-bit PWM	9	PWMx Configuration Register		40005898
16-bit PWM	9	TEST		4000589C
16-bit PWM	10	PWMx Counter ON Time Register		400058A0
16-bit PWM	10	PWMx Counter OFF Time Register		400058A4
16-bit PWM	10	PWMx Configuration Register		400058A8
16-bit PWM	10	TEST		400058AC

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
16-bit PWM	11	PWMx Counter ON Time Register		400058B0
16-bit PWM	11	PWMx Counter OFF Time Register		400058B4
16-bit PWM	11	PWMx Configuration Register		400058B8
16-bit PWM	11	TEST		400058BC
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
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
16-bit Tach	2	TACHx Control Register		40006020
16-bit Tach	2	TACHx Status Register		40006024
16-bit Tach	2	TACHx High Limit Register		40006028
16-bit Tach	2	TACHx Low Limit Register		4000602C
16-bit Tach	3	TACHx Control Register		40006030
16-bit Tach	3	TACHx Status Register		40006034
16-bit Tach	3	TACHx High Limit Register		40006038
16-bit Tach	3	TACHx Low Limit Register		4000603C
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 Channel 12 Reading Register		40007C44
ADC	0	ADC Channel 13 Reading Register		40007C48
ADC	0	ADC Channel 14 Reading Register		40007C4C

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ADC	0	ADC Channel 15 Reading Register		40007C50
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
RPM2PWM	0	Fan Setting Register		4000A000
RPM2PWM	0	Fan Configuration 1 Register		4000A002
RPM2PWM	0	Fan Configuration 2 Register		4000A003
RPM2PWM	0	PWM Divide Register		4000A004
RPM2PWM	0	Gain Register		4000A005
RPM2PWM	0	Fan Spin Up Configuration Register		4000A006
RPM2PWM	0	Fan Step Register		4000A007
RPM2PWM	0	Fan Minimum Drive Register		4000A008
RPM2PWM	0	Valid TACH Count Register		4000A009
RPM2PWM	0	Fan Drive Fail Band Register		4000A00A
RPM2PWM	0	TACH Target Register		4000A00C
RPM2PWM	0	TACH Reading Register		4000A00E
RPM2PWM	0	PWM Driver Base Frequency Register		4000A010
RPM2PWM	0	Fan Status Register		4000A011
RPM2PWM	0	TEST		4000A012
RPM2PWM	0	TEST		4000A014
RPM2PWM	0	TEST		4000A015
RPM2PWM	0	TEST		4000A016
RPM2PWM	0	TEST		4000A017
RPM2PWM	1	Fan Setting Register		4000A080
RPM2PWM	1	PWM Divide Register		4000A081
RPM2PWM	1	Fan Configuration 1 Register		4000A082
RPM2PWM	1	Fan Configuration 2 Register		4000A083
RPM2PWM	1	Reserved		4000A084
RPM2PWM	1	Gain Register		4000A085
RPM2PWM	1	Fan Spin Up Configuration Register		4000A086
RPM2PWM	1	Fan Step Register		4000A087
RPM2PWM	1	Fan Minimum Drive Register		4000A088
RPM2PWM	1	Valid TACH Count Register		4000A089
RPM2PWM	1	Fan Drive Fail Band Register		4000A08A

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
RPM2PWM	1	TACH Target Register		4000A08C
RPM2PWM	1	TACH Reading Register		4000A08E
RPM2PWM	1	PWM Driver Base Frequency Register		4000A090
RPM2PWM	1	Fan Status Register		4000A091
RPM2PWM	1	TEST		4000A092
RPM2PWM	1	TEST		4000A094
RPM2PWM	1	TEST		4000A095
RPM2PWM	1	TEST		4000A096
RPM2PWM	1	TEST		4000A097
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
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 Register Bank	0	Embedded Reset De-bounce Enable Register		4000A434
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
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
Blinking-Breathing PWM	2	LED Configuration Register		4000BA00
Blinking-Breathing PWM	2	LED Limits Register		4000BA04
Blinking-Breathing PWM	2	LED Delay Register		4000BA08

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Blinking-Breathing PWM	2	LED Update Stepsize Register		4000BA0C
Blinking-Breathing PWM	2	LED Update Interval Register		4000BA10
Blinking-Breathing PWM	2	LED Output Delay		4000BA14
Blinking-Breathing PWM	3	LED Configuration Register		4000BB00
Blinking-Breathing PWM	3	LED Limits Register		4000BB04
Blinking-Breathing PWM	3	LED Delay Register		4000BB08
Blinking-Breathing PWM	3	LED Update Stepsize Register		4000BB0C
Blinking-Breathing PWM	3	LED Update Interval Register		4000BB10
Blinking-Breathing PWM	3	LED Output Delay		4000BB14
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
EC Register Bank	0	TEST		4000FC24
EC Register Bank	0	WDT Event Count Register		4000FC28
EC Register Bank	0	AES HASH Byte Swap Control Register		4000FC2C
EC Register Bank	0	TEST		4000FC44
EC Register Bank	0	TEST		4000FC48
EC Register Bank	0	TEST		4000FC4C
EC Register Bank	0	TEST		4000FC54
EC Register Bank	0	TEST		4000FC5C
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	JTAG Master Configuration Register		4000FC70
EC Register Bank	0	JTAG Master Status Register		4000FC74
EC Register Bank	0	JTAG Master TDO Register		4000FC78
EC Register Bank	0	JTAG Master TDI Register		4000FC7C
EC Register Bank	0	JTAG Master TMS Register		4000FC80
EC Register Bank	0	JTAG Master Command Register		4000FC84
EC Register Bank	0	Vwire FW Override Register		4000FC90
EC Register Bank	0	Analog Comparator Control		4000FC94
EC Register Bank	0	Comparator Sleep Control		4000FC98
EC Register Bank	0	TEST		4000FCF0
EC Register Bank	0	TEST		4000FD00
EC Register Bank	0	JTAG Master Configuration Register		4000FD70
EC Register Bank	0	JTAG Master Status Register		4000FD74
EC Register Bank	0	JTAG Master TDO Register		4000FD78
EC Register Bank	0	JTAG Master TDI Register		4000FD7C
EC Register Bank	0	JTAG Master TMS Register		4000FD80
EC Register Bank	0	JTAG Master Command Register		4000FD84
EC Register Bank	0	TEST		4000FD88
EC Register Bank	0	Virtual Wire Source Configuration Register		4000FD90
EC Register Bank	0	Comparator Control Register		4000FD94
EC Register Bank	0	Comparator Sleep Control Register		4000FD98
Power Clocks and Resets	0	System Sleep Control Register		40080100
Power Clocks and Resets	0	Processor Clock Control Register		40080104

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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
Power Clocks and Resets	0	VBAT Soft Reset Register		40080188
Power Clocks and Resets	0	Source 32KHz Clock VTR Register		4008018C
Power Clocks and Resets	0	TEST		40080190
Power Clocks and Resets	0	Counter 32KHz Period Register		400801C0
Power Clocks and Resets	0	Counter 32KHz Pulse High Register		400801C4
Power Clocks and Resets	0	Counter 32KHz Period Minimum Register		400801C8
Power Clocks and Resets	0	Counter 32KHz Period Maximum Register		400801CC
Power Clocks and Resets	0	Counter 32KHz Duty Variation Register		400801D0
Power Clocks and Resets	0	Counter 32KHz Duty Variation Maximum Register		400801D4
Power Clocks and Resets	0	Counter 32KHz Valid Register		400801D8
Power Clocks and Resets	0	Counter 32KHz Valid Minimum Register		400801DC
Power Clocks and Resets	0	Counter 32KHz Control Register		400801E0
Power Clocks and Resets	0	Source 32KHz Interrupt Status Register		400801E4
Power Clocks and Resets	0	Source 32KHz Interrupt Enable Register		400801E8
GPIO	0	GPIO005 Pin Control Register		40081014
GPIO	0	GPIO006 Pin Control Register		40081018
GPIO	0	GPIO007 Pin Control Register		4008101C
GPIO	0	GPIO010 Pin Control Register		40081020
GPIO	0	GPIO012 Pin Control Register		40081028
GPIO	0	GPIO013 Pin Control Register		4008102C
GPIO	0	GPIO014 Pin Control Register		40081030

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO016 Pin Control Register		40081038
GPIO	0	GPIO017 Pin Control Register		4008103C
GPIO	0	GPIO022 Pin Control Register		40081048
GPIO	0	GPIO023 Pin Control Register		4008104C
GPIO	0	GPIO024 Pin Control Register		40081050
GPIO	0	GPIO031 Pin Control Register		40081064
GPIO	0	GPIO032 Pin Control Register		40081068
GPIO	0	GPIO035 Pin Control Register		40081074
GPIO	0	GPIO040 Pin Control Register		40081080
GPIO	0	GPIO050 Pin Control Register		400810A0
GPIO	0	GPIO051 Pin Control Register		400810A4
GPIO	0	GPIO053 Pin Control Register		400810AC
GPIO	0	GPIO054 Pin Control Register		400810B0
GPIO	0	GPIO057 Pin Control Register		400810BC
GPIO	0	GPIO067 Pin Control Register		400810DC
GPIO	0	GPIO104 Pin Control Register		40081110
GPIO	0	GPIO105 Pin Control Register		40081114
GPIO	0	GPIO106 Pin Control Register		40081118
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	GPIO130 Pin Control Register		40081160
GPIO	0	GPIO131 Pin Control Register		40081164
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	GPIO152 Pin Control Register		400811A8
GPIO	0	GPIO156 Pin Control Register		400811B8
GPIO	0	GPIO165 Pin Control Register		400811D4
GPIO	0	GPIO170 Pin Control Register		400811E0
GPIO	0	GPIO171 Pin Control Register		400811E4
GPIO	0	GPIO175 Pin Control Register		400811F4
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	GPIO205 Pin Control Register		40081214
GPIO	0	GPIO206 Pin Control Register		40081218
GPIO	0	GPIO207 Pin Control Register		4008121C

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO221 Pin Control Register		40081244
GPIO	0	GPIO224 Pin Control Register		40081250
GPIO	0	GPIO226 Pin Control Register		40081258
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	GPIO005 Pin Control2 Register		40081514
GPIO	0	GPIO006 Pin Control2 Register		40081518
GPIO	0	GPIO007 Pin Control2 Register		4008151C
GPIO	0	GPIO010 Pin Control2 Register		40081520
GPIO	0	GPIO012 Pin Control2 Register		40081528
GPIO	0	GPIO013 Pin Control2 Register		4008152C
GPIO	0	GPIO014 Pin Control2 Register		40081530
GPIO	0	GPIO016 Pin Control2 Register		40081538
GPIO	0	GPIO022 Pin Control2 Register		40081548
GPIO	0	GPIO023 Pin Control2 Register		4008154C
GPIO	0	GPIO024 Pin Control2 Register		40081550
GPIO	0	GPIO031 Pin Control2 Register		40081564
GPIO	0	GPIO032 Pin Control2 Register		40081568
GPIO	0	GPIO035 Pin Control2 Register		40081574
GPIO	0	GPIO040 Pin Control2 Register		40081580
GPIO	0	GPIO050 Pin Control2 Register		400815A0
GPIO	0	GPIO051 Pin Control2 Register		400815A4
GPIO	0	GPIO053 Pin Control2 Register		400815AC
GPIO	0	GPIO054 Pin Control2 Register		400815B0
GPIO	0	GPIO057 Pin Control2 Register		400815BC
GPIO	0	GPIO067 Pin Control2 Register		400815DC
GPIO	0	GPIO104 Pin Control2 Register		40081610
GPIO	0	GPIO105 Pin Control2 Register		40081614
GPIO	0	GPIO106 Pin Control2 Register		40081618
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO126 Pin Control2 Register		40081658
GPIO	0	GPIO130 Pin Control2 Register		40081660
GPIO	0	GPIO131 Pin Control2 Register		40081664
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	GPIO152 Pin Control2 Register		400816A8
GPIO	0	GPIO156 Pin Control2 Register		400816B8
GPIO	0	GPIO165 Pin Control2 Register		400816D4
GPIO	0	GPIO170 Pin Control2 Register		400816E0
GPIO	0	GPIO171 Pin Control2 Register		400816E4
GPIO	0	GPIO175 Pin Control2 Register		400816F4
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	GPIO205 Pin Control2 Register		40081714
GPIO	0	GPIO206 Pin Control2 Register		40081718
GPIO	0	GPIO207 Pin Control2 Register		4008171C
GPIO	0	GPIO221 Pin Control2 Register		40081744
GPIO	0	GPIO224 Pin Control2 Register		40081750
GPIO	0	GPIO226 Pin Control2 Register		40081758
OTP	0	Write Lock Register		40082044
OTP	0	Read Lock Register		40082048
OTP	0	Write Byte Lock Register		4008204C
OTP	0	Read Byte Lock Register		40082050
Mailbox	0	MBX_INDEX Register		400F0000
Mailbox	0	MBX_DATA Register		400F0001
Mailbox	0	HOST-to-EC Mailbox Register		400F0100
Mailbox	0	EC-to-Host Mailbox Register		400F0104
Mailbox	0	SMI Interrupt Source Register		400F0108
Mailbox	0	SMI Interrupt Mask Register		400F010C
Mailbox	0	Mailbox register [3:0]		400F0110
Mailbox	0	Mailbox register [7:4]		400F0114
Mailbox	0	Mailbox register [B:8]		400F0118
Mailbox	0	Mailbox register [F:C]		400F011C
Mailbox	0	Mailbox register [13:10]		400F0120
Mailbox	0	Mailbox register [17:14]		400F0124
Mailbox	0	Mailbox register [1B:18]		400F0128
Mailbox	0	Mailbox register [1F:1C]		400F012C
ACPI EC Channel	0	ACPI OS Data Register Byte 0 Register	Run-time	400F0800

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ACPI EC Channel	0	ACPI OS Data Register Byte 1 Register	Run-time	400F0801
ACPI EC Channel	0	ACPI OS Data Register Byte 2 Register	Run-time	400F0802
ACPI EC Channel	0	ACPI OS Data Register Byte 3 Register	Run-time	400F0803
ACPI EC Channel	0	ACPI OS COMMAND Register	Run-time	400F0804
ACPI EC Channel	0	OS STATUS OS Register	Run-time	400F0804
ACPI EC Channel	0	OS Byte Control Register	Run-time	400F0805
ACPI EC Channel	0	Reserved	Run-time	400F0806
ACPI EC Channel	0	Reserved	Run-time	400F0807
ACPI EC Channel	0	EC2OS Data EC Byte 0 Register		400F0900
ACPI EC Channel	0	EC2OS Data EC Byte 1 Register		400F0901
ACPI EC Channel	0	EC2OS Data EC Byte 2 Register		400F0902
ACPI EC Channel	0	EC2OS Data EC Byte 3 Register		400F0903
ACPI EC Channel	0	EC STATUS Register		400F0904
ACPI EC Channel	0	EC Byte Control Register		400F0905
ACPI EC Channel	0	Reserved		400F0906
ACPI EC Channel	0	Reserved		400F0907
ACPI EC Channel	0	OS2EC Data EC Byte 0 Register		400F0908
ACPI EC Channel	0	OS2EC Data EC Byte 1 Register		400F0909
ACPI EC Channel	0	OS2EC Data EC Byte 2 Register		400F090A
ACPI EC Channel	0	OS2EC Data EC Byte 3 Register		400F090B
ACPI EC Channel	1	ACPI OS Data Register Byte 0 Register	Run-time	400F0C00
ACPI EC Channel	1	ACPI OS Data Register Byte 1 Register	Run-time	400F0C01
ACPI EC Channel	1	ACPI OS Data Register Byte 2 Register	Run-time	400F0C02
ACPI EC Channel	1	ACPI OS Data Register Byte 3 Register	Run-time	400F0C03
ACPI EC Channel	1	ACPI OS COMMAND Register	Run-time	400F0C04
ACPI EC Channel	1	OS STATUS OS Register	Run-time	400F0C04
ACPI EC Channel	1	OS Byte Control Register	Run-time	400F0C05
ACPI EC Channel	1	Reserved	Run-time	400F0C06
ACPI EC Channel	1	Reserved	Run-time	400F0C07
ACPI EC Channel	1	EC2OS Data EC Byte 0 Register		400F0D00

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ACPI EC Channel	1	EC2OS Data EC Byte 1 Register		400F0D01
ACPI EC Channel	1	EC2OS Data EC Byte 2 Register		400F0D02
ACPI EC Channel	1	EC2OS Data EC Byte 3 Register		400F0D03
ACPI EC Channel	1	EC STATUS Register		400F0D04
ACPI EC Channel	1	EC Byte Control Register		400F0D05
ACPI EC Channel	1	Reserved		400F0D06
ACPI EC Channel	1	Reserved		400F0D07
ACPI EC Channel	1	OS2EC Data EC Byte 0 Register		400F0D08
ACPI EC Channel	1	OS2EC Data EC Byte 1 Register		400F0D09
ACPI EC Channel	1	OS2EC Data EC Byte 2 Register		400F0D0A
ACPI EC Channel	1	OS2EC Data EC Byte 3 Register		400F0D0B
ACPI EC Channel	2	ACPI OS Data Register Byte 0 Register	Run-time	400F1000
ACPI EC Channel	2	ACPI OS Data Register Byte 1 Register	Run-time	400F1001
ACPI EC Channel	2	ACPI OS Data Register Byte 2 Register	Run-time	400F1002
ACPI EC Channel	2	ACPI OS Data Register Byte 3 Register	Run-time	400F1003
ACPI EC Channel	2	ACPI OS COMMAND Register	Run-time	400F1004
ACPI EC Channel	2	OS STATUS OS Register	Run-time	400F1004
ACPI EC Channel	2	OS Byte Control Register	Run-time	400F1005
ACPI EC Channel	2	Reserved	Run-time	400F1006
ACPI EC Channel	2	Reserved	Run-time	400F1007
ACPI EC Channel	2	EC2OS Data EC Byte 0 Register		400F1100
ACPI EC Channel	2	EC2OS Data EC Byte 1 Register		400F1101
ACPI EC Channel	2	EC2OS Data EC Byte 2 Register		400F1102
ACPI EC Channel	2	EC2OS Data EC Byte 3 Register		400F1103
ACPI EC Channel	2	EC STATUS Register		400F1104
ACPI EC Channel	2	EC Byte Control Register		400F1105
ACPI EC Channel	2	Reserved		400F1106
ACPI EC Channel	2	Reserved		400F1107
ACPI EC Channel	2	OS2EC Data EC Byte 0 Register		400F1108
ACPI EC Channel	2	OS2EC Data EC Byte 1 Register		400F1109
ACPI EC Channel	2	OS2EC Data EC Byte 2 Register		400F110A
ACPI EC Channel	2	OS2EC Data EC Byte 3 Register		400F110B
ACPI EC Channel	3	ACPI OS Data Register Byte 0 Register	Run-time	400F1400
ACPI EC Channel	3	ACPI OS Data Register Byte 1 Register	Run-time	400F1401

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ACPI EC Channel	3	ACPI OS Data Register Byte 2 Register	Run-time	400F1402
ACPI EC Channel	3	ACPI OS Data Register Byte 3 Register	Run-time	400F1403
ACPI EC Channel	3	ACPI OS COMMAND Register	Run-time	400F1404
ACPI EC Channel	3	OS STATUS OS Register	Run-time	400F1404
ACPI EC Channel	3	OS Byte Control Register	Run-time	400F1405
ACPI EC Channel	3	Reserved	Run-time	400F1406
ACPI EC Channel	3	Reserved	Run-time	400F1407
ACPI EC Channel	3	EC2OS Data EC Byte 0 Register		400F1500
ACPI EC Channel	3	EC2OS Data EC Byte 1 Register		400F1501
ACPI EC Channel	3	EC2OS Data EC Byte 2 Register		400F1502
ACPI EC Channel	3	EC2OS Data EC Byte 3 Register		400F1503
ACPI EC Channel	3	EC STATUS Register		400F1504
ACPI EC Channel	3	EC Byte Control Register		400F1505
ACPI EC Channel	3	Reserved		400F1506
ACPI EC Channel	3	Reserved		400F1507
ACPI EC Channel	3	OS2EC Data EC Byte 0 Register		400F1508
ACPI EC Channel	3	OS2EC Data EC Byte 1 Register		400F1509
ACPI EC Channel	3	OS2EC Data EC Byte 2 Register		400F150A
ACPI EC Channel	3	OS2EC Data EC Byte 3 Register		400F150B
ACPI EC Channel	4	ACPI OS Data Register Byte 0 Register	Run-time	400F1800
ACPI EC Channel	4	ACPI OS Data Register Byte 1 Register	Run-time	400F1801
ACPI EC Channel	4	ACPI OS Data Register Byte 2 Register	Run-time	400F1802
ACPI EC Channel	4	ACPI OS Data Register Byte 3 Register	Run-time	400F1803
ACPI EC Channel	4	ACPI OS COMMAND Register	Run-time	400F1804
ACPI EC Channel	4	OS STATUS OS Register	Run-time	400F1804
ACPI EC Channel	4	OS Byte Control Register	Run-time	400F1805
ACPI EC Channel	4	Reserved	Run-time	400F1806
ACPI EC Channel	4	Reserved	Run-time	400F1807
ACPI EC Channel	4	EC2OS Data EC Byte 0 Register		400F1900
ACPI EC Channel	4	EC2OS Data EC Byte 1 Register		400F1901
ACPI EC Channel	4	EC2OS Data EC Byte 2 Register		400F1902

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ACPI EC Channel	4	EC2OS Data EC Byte 3 Register		400F1903
ACPI EC Channel	4	EC STATUS Register		400F1904
ACPI EC Channel	4	EC Byte Control Register		400F1905
ACPI EC Channel	4	Reserved		400F1906
ACPI EC Channel	4	Reserved		400F1907
ACPI EC Channel	4	OS2EC Data EC Byte 0 Register		400F1908
ACPI EC Channel	4	OS2EC Data EC Byte 1 Register		400F1909
ACPI EC Channel	4	OS2EC Data EC Byte 2 Register		400F190A
ACPI EC Channel	4	OS2EC Data EC Byte 3 Register		400F190B
ACPI PM1	0	Power Management 1 Status 1 Register	Run-time	400F1C00
ACPI PM1	0	Power Management 1 Status 2 Register	Run-time	400F1C01
ACPI PM1	0	Power Management 1 Enable 1 Register	Run-time	400F1C02
ACPI PM1	0	Power Management 1 Enable 2 Register	Run-time	400F1C03
ACPI PM1	0	Power Management 1 Control 1 Register	Run-time	400F1C04
ACPI PM1	0	Power Management 1 Control 2 Register	Run-time	400F1C05
ACPI PM1	0	Power Management 2 Control 1 Register	Run-time	400F1C06
ACPI PM1	0	Power Management 2 Control 2 Register	Run-time	400F1C07
ACPI PM1	0	Power Management 1 Status 1 Register		400F1D00
ACPI PM1	0	Power Management 1 Status 2 Register		400F1D01
ACPI PM1	0	Power Management 1 Enable 1 Register		400F1D02
ACPI PM1	0	Power Management 1 Enable 2 Register		400F1D03
ACPI PM1	0	Power Management 1 Control 1 Register		400F1D04
ACPI PM1	0	Power Management 1 Control 2 Register		400F1D05
ACPI PM1	0	Power Management 2 Control 1 Register		400F1D06
ACPI PM1	0	Power Management 2 Control 2 Register		400F1D07
ACPI PM1	0	EC_PM_STS Register		400F1D10
Port92-Legacy	0	Port 92 Register	Run-time	400F2000
Port92-Legacy	0	GATEA20 Control Register		400F2100
Port92-Legacy	0	SETGA20L Register		400F2108
Port92-Legacy	0	RSTGA20L Register		400F210C
Port92-Legacy	0	Port 92 Enable	Con-fig	400F2330
UART	0	Receive Buffer Register	Run-time	400F2400
UART	0	Transmit Buffer Register	Run-time	400F2400

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
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	Con-fig	400F2730
UART	0	Configuration Select Register	Con-fig	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
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

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
UART	1	Activate Register	Con-fig	400F2B30
UART	1	Configuration Select Register	Con-fig	400F2BF0
eSPI SAF Bridge Compo-nent	0	Test		40008000
eSPI SAF Bridge Compo-nent	0	SAF EC Portal Command Register		40008018
eSPI SAF Bridge Compo-nent	0	SAF EC Portal Flash Address Register		4000801C
eSPI SAF Bridge Compo-nent	0	SAF EC Portal Start Register		40008020
eSPI SAF Bridge Compo-nent	0	SAF EC Portal Buffer Address Register		40008024
eSPI SAF Bridge Compo-nent	0	SAF EC Portal Status Register		40008028
eSPI SAF Bridge Compo-nent	0	SAF EC Portal Interrupt Enable Register		4000802C
eSPI SAF Bridge Compo-nent	0	SAF Flash Configuration Size Limit Register		40008030
eSPI SAF Bridge Compo-nent	0	SAF Flash Configuration Threshold Register		40008034
eSPI SAF Bridge Compo-nent	0	SAF Flash Configuration Misc Register		40008038
eSPI SAF Bridge Compo-nent	0	SAF eSPI Monitor Status Register		4000803C
eSPI SAF Bridge Compo-nent	0	SAF eSPI Monitor Interrupt Enable Register		40008040
eSPI SAF Bridge Compo-nent	0	SAF EC Busy Register		40008044
eSPI SAF Bridge Compo-nent	0	TEST		40008048
eSPI SAF Bridge Compo-nent	0	CS0 Opcode:SAF Flash Configuration Opcode Register A		4000804C
eSPI SAF Bridge Compo-nent	0	CS0 Opcode:SAF Flash Configuration Opcode Register B		40008050
eSPI SAF Bridge Compo-nent	0	CS0 opcode:SAF Flash Configuration Opcode Register C		40008054
eSPI SAF Bridge Compo-nent	0	CS0 Opcode;SAF Flash Configuration Per-Flash Descriptors Register		40008058
eSPI SAF Bridge Compo-nent	0	CS1 Opcode:SAF Flash Configuration Opcode Register A		4000805C
eSPI SAF Bridge Compo-nent	0	CS1 Opcode:SAF Flash Configuration Opcode Register B		40008060
eSPI SAF Bridge Compo-nent	0	CS1 opcode:SAF Flash Configuration Opcode Register C		40008064
eSPI SAF Bridge Compo-nent	0	CS1 Opcode;SAF Flash Configuration Per-Flash Descriptors Register		40008068

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF Flash Configuration General Descriptors Register		4000806C
eSPI SAF Bridge Component	0	SAF Protection Lock Bit Register		40008070
eSPI SAF Bridge Component	0	SAF Protection Dirty Bit Register		40008074
eSPI SAF Bridge Component	0	SAF Tag Map Register 0		40008078
eSPI SAF Bridge Component	0	SAF Tag Map Register 1		4000807C
eSPI SAF Bridge Component	0	SAF Tag Map Register 2		40008080
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008084
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008088
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000808C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008090
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008094
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008098
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000809C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080A0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080A4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080A8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080AC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080B0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080B4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080B8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080BC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080C0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080C4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080C8

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080CC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080D0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080D4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080D8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080DC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080E0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080E4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080E8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080EC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080F0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080F4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080F8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080FC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008100
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008104
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008108
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000810C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008110
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008114
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008118
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000811C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008120
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008124
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008128

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000812C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008130
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008134
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008138
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000813C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008140
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008144
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008148
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000814C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008150
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008154
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008158
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000815C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008160
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008164
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008168
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000816C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008170
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008174
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008178
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000817C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008180
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008184
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008188

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000818C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008190
eSPI SAF Bridge Component	0	SAF Poll Timeout Register		40008194
eSPI SAF Bridge Component	0	SAF Poll Interval Register		40008198
eSPI SAF Bridge Component	0	SAF Suspend/Resume Interval Register		4000819C
eSPI SAF Bridge Component	0	SAF Consecutive Read Timeout Register		400081A0
eSPI SAF Bridge Component	0	SAF Flash Configuration Poll2 Mask Register		400081A4
eSPI SAF Bridge Component	0	SAF Flash Configuration Special Mode Register		400081A8
eSPI SAF Bridge Component	0	SAF Suspend Check Delay Register		400081AC
eSPI SAF Bridge Component	0	SAF Flash Configuration Special Mode Register		400081B0
eSPI SAF Bridge Component	0	SAF DnX Protection Bypass		400081B4
eSPI SAF Bridge Component	0	SAF Activity Count Reload Value Register		400081B8
eSPI SAF Bridge Component	0	SAF Power Down Control Register		400081BC
eSPI SAF Bridge Component	0	SAF Memory Power Status Register		400081C0
eSPI SAF Bridge Component	0	SAF Config CS0 Opcode Register		400081C4
eSPI SAF Bridge Component	0	SAF Config CS1 Opcode Register		400081C8
eSPI SAF Bridge Component	0	SAF Flash Power Down /Up Timeout Register		400081CC
eSPI SAF Bridge Component	0	Clock Divider for CS0 Register		40008200
eSPI SAF Bridge Component	0	Clock Divider for CS1 Register		40008204
eSPI SAF Bridge Component	0	SAF RPMC OP2 eSPI Result Register		40008208
eSPI SAF Bridge Component	0	SAF RPMC OP2 EC0 Result Register		4000820C
eSPI SAF Bridge Component	0	SAF RPMC OP2 EC1 Result Register		40008210
eSPI SAF Communication Registers	0	SAF Communication Mode Register		400712B8
EMI	0	HOST-to-EC Mailbox Register	Run-time	400F4000

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
EMI	0	EC-to-HOST Mailbox Register	Run-time	400F4001
EMI	0	EC Address LSB Register	Run-time	400F4002
EMI	0	EC Address MSB Register	Run-time	400F4003
EMI	0	EC Data Byte 0 Register	Run-time	400F4004
EMI	0	EC Data Byte 1 Register	Run-time	400F4005
EMI	0	EC Data Byte 2 Register	Run-time	400F4006
EMI	0	EC Data Byte 3 Register	Run-time	400F4007
EMI	0	Interrupt Source LSB Register	Run-time	400F4008
EMI	0	Interrupt Source MSB Register	Run-time	400F4009
EMI	0	Interrupt Mask LSB Register	Run-time	400F400A
EMI	0	Interrupt Mask MSB Register	Run-time	400F400B
EMI	0	Application ID Register	Run-time	400F400C
EMI	0	Application ID Assignment Register	Run-time	400F4010
EMI	0	HOST-to-EC Mailbox Register		400F4100
EMI	0	EC-to-HOST Mailbox Register		400F4101
EMI	0	Memory Base Address 0 Register		400F4104
EMI	0	Memory Read Limit 0 Register		400F4108
EMI	0	Memory Write Limit 0 Register		400F410A
EMI	0	Memory Base Address 1 Register		400F410C
EMI	0	Memory Read Limit 1 Register		400F4110
EMI	0	Memory Write Limit 1 Register		400F4112
EMI	0	Interrupt Set Register		400F4114
EMI	0	Host Clear Enable Register		400F4116
EMI	0	Application ID Status 0 Register		400F4120
EMI	0	Application ID Status 1 Register		400F4124
EMI	0	Application ID Status 2 Register		400F4128
EMI	0	Application ID Status 3 Register		400F412C
EMI	0	Application ID Status 4 Register		400F4130
EMI	0	Application ID Status 5 Register		400F4134
EMI	0	Application ID Status 6 Register		400F4138
EMI	0	Application ID Status 7 Register		400F413C
EMI	1	HOST-to-EC Mailbox Register	Run-time	400F4400

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
EMI	1	EC-to-HOST Mailbox Register	Run-time	400F4401
EMI	1	EC Address LSB Register	Run-time	400F4402
EMI	1	EC Address MSB Register	Run-time	400F4403
EMI	1	EC Data Byte 0 Register	Run-time	400F4404
EMI	1	EC Data Byte 1 Register	Run-time	400F4405
EMI	1	EC Data Byte 2 Register	Run-time	400F4406
EMI	1	EC Data Byte 3 Register	Run-time	400F4407
EMI	1	Interrupt Source LSB Register	Run-time	400F4408
EMI	1	Interrupt Source MSB Register	Run-time	400F4409
EMI	1	Interrupt Mask LSB Register	Run-time	400F440A
EMI	1	Interrupt Mask MSB Register	Run-time	400F440B
EMI	1	Application ID Register	Run-time	400F440C
EMI	1	Application ID Assignment Register	Run-time	400F4410
EMI	1	HOST-to-EC Mailbox Register		400F4500
EMI	1	EC-to-HOST Mailbox Register		400F4501
EMI	1	Memory Base Address 0 Register		400F4504
EMI	1	Memory Read Limit 0 Register		400F4508
EMI	1	Memory Write Limit 0 Register		400F450A
EMI	1	Memory Base Address 1 Register		400F450C
EMI	1	Memory Read Limit 1 Register		400F4510
EMI	1	Memory Write Limit 1 Register		400F4512
EMI	1	Interrupt Set Register		400F4514
EMI	1	Host Clear Enable Register		400F4516
EMI	1	Application ID Status 0 Register		400F4520
EMI	1	Application ID Status 1 Register		400F4524
EMI	1	Application ID Status 2 Register		400F4528
EMI	1	Application ID Status 3 Register		400F452C
EMI	1	Application ID Status 4 Register		400F4530
EMI	1	Application ID Status 5 Register		400F4534
EMI	1	Application ID Status 6 Register		400F4538
EMI	1	Application ID Status 7 Register		400F453C
EMI	2	HOST-to-EC Mailbox Register	Run-time	400F4800

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
EMI	2	EC-to-HOST Mailbox Register	Run-time	400F4801
EMI	2	EC Address LSB Register	Run-time	400F4802
EMI	2	EC Address MSB Register	Run-time	400F4803
EMI	2	EC Data Byte 0 Register	Run-time	400F4804
EMI	2	EC Data Byte 1 Register	Run-time	400F4805
EMI	2	EC Data Byte 2 Register	Run-time	400F4806
EMI	2	EC Data Byte 3 Register	Run-time	400F4807
EMI	2	Interrupt Source LSB Register	Run-time	400F4808
EMI	2	Interrupt Source MSB Register	Run-time	400F4809
EMI	2	Interrupt Mask LSB Register	Run-time	400F480A
EMI	2	Interrupt Mask MSB Register	Run-time	400F480B
EMI	2	Application ID Register	Run-time	400F480C
EMI	2	Application ID Assignment Register	Run-time	400F4810
EMI	2	HOST-to-EC Mailbox Register		400F4900
EMI	2	EC-to-HOST Mailbox Register		400F4901
EMI	2	Memory Base Address 0 Register		400F4904
EMI	2	Memory Read Limit 0 Register		400F4908
EMI	2	Memory Write Limit 0 Register		400F490A
EMI	2	Memory Base Address 1 Register		400F490C
EMI	2	Memory Read Limit 1 Register		400F4910
EMI	2	Memory Write Limit 1 Register		400F4912
EMI	2	Interrupt Set Register		400F4914
EMI	2	Host Clear Enable Register		400F4916
EMI	2	Application ID Status 0 Register		400F4920
EMI	2	Application ID Status 1 Register		400F4924
EMI	2	Application ID Status 2 Register		400F4928
EMI	2	Application ID Status 3 Register		400F492C
EMI	2	Application ID Status 4 Register		400F4930
EMI	2	Application ID Status 5 Register		400F4934
EMI	2	Application ID Status 6 Register		400F4938
EMI	2	Application ID Status 7 Register		400F493C
Real Time Clock	0	Seconds Register	Run-time	400F5000

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Real Time Clock	0	Seconds Alarm Register	Run-time	400F5001
Real Time Clock	0	Minutes Register	Run-time	400F5002
Real Time Clock	0	Minutes Alarm Register	Run-time	400F5003
Real Time Clock	0	Hours Register	Run-time	400F5004
Real Time Clock	0	Hours Alarm Register	Run-time	400F5005
Real Time Clock	0	Day of Week Register	Run-time	400F5006
Real Time Clock	0	Day of Month Register	Run-time	400F5007
Real Time Clock	0	Month Register	Run-time	400F5008
Real Time Clock	0	Year Register	Run-time	400F5009
Real Time Clock	0	Register A	Run-time	400F500A
Real Time Clock	0	Register B	Run-time	400F500B
Real Time Clock	0	Register C	Run-time	400F500C
Real Time Clock	0	Register D	Run-time	400F500D
Real Time Clock	0	Reserved	Run-time	400F500E
Real Time Clock	0	Reserved	Run-time	400F500F
Real Time Clock	0	RTC Control Register	Run-time	400F5010
Real Time Clock	0	Week Alarm Register	Run-time	400F5014
Real Time Clock	0	Daylight Savings Forward Register	Run-time	400F5018
Real Time Clock	0	Daylight Savings Backward Register	Run-time	400F501C
Real Time Clock	0	TEST	Run-time	400F5020
32-Bit BIOS Debug Port (Port 80) Base	0	Host Data Register	Run-time	400F8000
32-Bit BIOS Debug Port (Port 80) Base	0	EC Data Register		400F8100
32-Bit BIOS Debug Port (Port 80) Base	0	EC Data Attribute Register		400F8101
32-Bit BIOS Debug Port (Port 80) Base	0	Configuration Register		400F8104

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
32-Bit BIOS Debug Port (Port 80) Base	0	Status Register		400F8108
32-Bit BIOS Debug Port (Port 80) Base	0	Interrupt Enable Register		400F8109
32-Bit BIOS Debug Port (Port 80) Base	0	Snap Short Register		400F810C
32-Bit BIOS Debug Port (Port 80) Base	0	Capture Register		400F8110
32-Bit BIOS Debug Port (Port 80) Base	0	Activate Register	Con-fig	400F8330
32-Bit BIOS Debug Port (Port 80) Alias	0	Host Data Register	Run-time	400F8400
32-Bit BIOS Debug Port (Port 80) Alias	0	Alias Activate Register	Con-fig	400F8730
32-Bit BIOS Debug Port (Port 80) Alias	0	Alias Byte Lane Register	Con-fig	400F87F0
Global Configuration	0	Global Configuration Reserved	Run-time	400FFF00
Global Configuration	0	Control	Run-time	400FFF02
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	OTP ID	Run-time	400FFF24
Global Configuration	0	Validation ID	Run-time	400FFF25
Global Configuration	0	Boot ROM Revision ID[15:0]	Run-time	400FFF26
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

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Global Configuration	0	TEST	Run-time	400FFF2D
Global Configuration	0	TEST	Run-time	400FFF2E
Global Configuration	0	TEST	Run-time	400FFF2F
ARM M4F	0	Auxiliary_Control		E000E008
ARM M4F	0	SystemTick_Ctrl_Status		E000E010
ARM M4F	0	SystemTick_Reload_Value		E000E014
ARM M4F	0	SystemTick_Current_Value		E000E018
ARM M4F	0	SystemTick_Calibration_Value		E000E01C
ARM M4F	0	CPU_ID		E000ED00
ARM M4F	0	Interrupt_Ctl_and_State		E000ED04
ARM M4F	0	Vector_Table_Offset		E000ED08
ARM M4F	0	Application_Interrupt_and_Reset_Ctl		E000ED0C
ARM M4F	0	System_Ctl		E000ED10
ARM M4F	0	Config_and_Ctl		E000ED14
ARM M4F	0	System_Handler_Priority1		E000ED18
ARM M4F	0	System_Handler_Priority2		E000ED1C
ARM M4F	0	System_Handler_Priority3		E000ED20
ARM M4F	0	System_Handler_Ctl_and_State		E000ED24
ARM M4F	0	Configurable_Fault_Status		E000ED28
ARM M4F	0	Hard_Fault_Status		E000ED2C
ARM M4F	0	Debug_Fault_Status		E000ED30
ARM M4F	0	Debug_Halting_Ctl_and_Status		E000EDF0
ARM M4F	0	Debug_Core_Register_Selector		E000EDF4
ARM M4F	0	Debug_Core_Register_Data		E000EDF8
ARM M4F	0	Debug_Exception_and_Monitor_Ctl		E000EDFC
ARM M4F	0	Bus_Fault_Address		E000ED38
ARM M4F	0	Auxiliary_Fault_Status		E000ED3C
ARM M4F	0	Processor_Feature0		E000ED40
ARM M4F	0	Processor_Feature1		E000ED44
ARM M4F	0	Debug_Features0		E000ED48
ARM M4F	0	Auxiliary_Features0		E000ED4C
ARM M4F	0	Memory_Model_Feature0		E000ED50
ARM M4F	0	Memory_Model_Feature1		E000ED54
ARM M4F	0	Memory_Model_Feature2		E000ED58
ARM M4F	0	Memory_Model_Feature3		E000ED5C
ARM M4F	0	Instruction_Set_Attributes0		E000ED60
ARM M4F	0	Instruction_Set_Attributes1		E000ED64
ARM M4F	0	Instruction_Set_Attributes2		E000ED68
ARM M4F	0	Instruction_Set_Attributes3		E000ED6C
ARM M4F	0	Instruction_Set_Attributes4		E000ED70
ARM M4F	0	Coprocessor_Access_Ctl		E000ED88

© 2021 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
ARM M4F	0	Software_Triggered_Interrupt		E000EF00
SPI Slave	0	SPI Communication Configuration Register		40007000
SPI Slave	0	SPI Slave Status Register		40007004
SPI Slave	0	SPI EC Status Register		40007008
SPI Slave	0	SPI Interrupt Enable Register		4000700C
SPI Slave	0	EC Interrupt Enable Register		40007010
SPI Slave	0	Memory Configuration Register		40007014
SPI Slave	0	Memory Base Address0 Register		40007018
SPI Slave	0	Memory Write Limit0 Register		4000701C
SPI Slave	0	Memory Read Limit0 Register		40007020
SPI Slave	0	Memory Base Address1 Register		40007024
SPI Slave	0	Memory Write Limit1 Register		40007028
SPI Slave	0	Memory Read Limit1 Register		4000702C
SPI Slave	0	RX FIFO Host BAR		40007030
SPI Slave	0	RX FIFO Byte CNTR		40007034
SPI Slave	0	TX FIFO Host BAR		40007038
SPI Slave	0	RX FIFO Byte CNTR		4000703C
SPI Slave	0	System Configuration Register		40007040
SPI Slave	0	SPI Master-to-EC Mailbox Register		40007044
SPI Slave	0	EC-to-SPI Master Mailbox Register		40007048
SPI Slave	0	Test Modes Register		4000704C

© 2021 Microchip Technology Inc.

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.	Pin Interface
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
VTR3	1.8V	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. See Note 4 for details.
VSSx	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: VBAT is connected to VTR1 in this package.

© 2021 Microchip Technology Inc.

4.4.1 I/O RAIL REQUIREMENTS

All pins are powered by four power supply pins: VTR1 and VTR2 . The VTR1 is fixed 3.3V and VTR2 pins may be connected to either a 3.3V or a 1.8V power supply.

After RESET_SYS, when the VTR2 power rail is stable, the IO pads connected to VTR2 power rail, auto-detect the IO voltage they are connected to. No software intervention is required.

4.4.2 VOLTAGE REFERENCES

Table 4-2 lists the External Voltage References to which the EEC1727 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 bit 0					

4.4.3 SYSTEM POWER SEQUENCING

TABLE 4-3: POWER GOOD SIGNAL DEFINITIONS

Power Good Signal	Description	Source
VCC_PWRGD	VCC_PWRGD is an input signal used to indicate when the main system power rail voltage is on and stable.	Pin Interface

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

TABLE 4-4: 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

TABLE 4-5: POWER SEQUENCING SIGNALS

Power Good Signal	Description	Source
SYSPWR_VALID	SYSPWR_VALID is an input used to indicated that system power is within operational range. This signal is used to detect surprise shutdown event. This feature is disabled by default and may be enabled through OTP bit selection. low = system power not valid high = system power valid	GPIO155 pin. There is no special hardware associated with this signal

TABLE 4-5: POWER SEQUENCING SIGNALS (CONTINUED)

Power Good Signal	Description	Source
DPWREN	DSW Power Regulator Enable is an open drain output signal. This signal is driven low if a Surprise Power Down event is detected. low = DSW Power Regulator off high = DSW Power Regulator on	There is no special hardware associated with this signal. The GPIO on which this output will be driven is configurable in the OTP. Any GPIO can be configured for this purpose.
DSW_PWRGD	DSW Power Regulator Good is an input used to inform EC that the DSW Power Regulator voltage is within operational range. low = DSW Power Regulator output not valid high = DSW Power Regulator output valid	There is no special hardware associated with this signal. The GPIO on which this output will be driven is configurable in the OTP. Any GPIO can be configured for this purpose.
DSW_PWROK	DSW Power OK is an open drain output signal that indicates components on this rail can be released from reset. This signal is driven low if a Surprise Power Down event is detected. low = System components powered by DSW must be in reset high = System components powered by DSW may be released from reset	There is no special hardware associated with this signal.
SLP_SUS#	SLP_SUS# is an input used to notify system when suspend power (i.e., Primary Rails) must be powered on or may be removed. This feature is disabled by default and may be enabled through OTP bit selection. low = Primary Rails are not ready to be powered on high = Primary Rails must be powered on	There is no special hardware associated with this signal.
SUS_PWR_EN	Primary Power Regulator Enable an open drain output signal that may be used to enable the Primary Power Regulator. This feature is disabled by default and may be enabled through OTP bit selection. This signal is driven low if a Surprise Power Down event is detected low = Primary Power Regulator off high = Primary Power Regulator on	There is no special hardware associated with this signal. The GPIO on which this output will be driven is configurable in the OTP. Any GPIO can be configured for this purpose.
PRIM_PWRGD	PRIM_PWRGD is an input signal that indicates that at least one primary rail is powered. For shared flash applications, it also indicates when the SPI Flash is powered and ready for EC access.	There is no special hardware associated with this signal.
RSMRST#	RSMRST# is an open drain output that indicates all Primary power rails are valid and devices may be released from reset. This signal is driven low if a Surprise Power Down event is detected low = System components powered by Primary power rails must be in reset high = System components powered by Primary power rails may be released from reset	There is no special hardware associated with this signal.

© 2021 Microchip Technology Inc.

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 in the chip.

TABLE 4-6: 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 VBAT SOURCE 32kHZ Register.	Pin Interface (XTAL1 and XTAL2) When used in singled-ended configuration, pin XTAL2 should be tied to the clock source and XTAL1 should be grounded.
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 VTR1.
60 MHz Ring Oscillator	32MHz	The 60MHz Ring Oscillator is used to supply a clock for the 96MHz main clock domain while the 96MHz PLL is not locked. Its frequency can range from 32Mhz to 92MHz.	Powered by VTR_CORE.
96 MHz	96MHz	The 96 MHz Phase Locked Loop generates a 96 MHz clock locked to the 32KHz Clock Source	Powered by VTR_CORE. May be stopped by Chip Power Management Features.
48MHz	48MHz	The 48MHz clock is derived from the 96MHz	
SPI Clock	1MHz - 66MHz	This clock is used only in the SPI Slave interface	External SPI Master

4.5.2 CLOCK DOMAINS

TABLE 4-7: CLOCK DOMAIN DEFINITIONS

Clock Domain	Description
32KHz	The clock source used as reference for PLL lock and System Clock controls.
32KHz Core	The clock source used by internal blocks that require an always-on low speed clock
96MHz	The clock source used for system clock controls for divide down PLL or Dumb Ring Oscillator.
2MHz	Internally generated 2 MHz clock from 96MHz clock.
100KHz	A low-speed clock derived from the 48MHz clock domain. Used as a time base for PWMsand Tachs.
EC_CLK	The clock used by the EC processor. The frequency is determined by the Processor Clock Control Register.

TABLE 4-7: CLOCK DOMAIN DEFINITIONS (CONTINUED)

Clock Domain	Description
MCLK	The clock used by the Individual blocks. This can be 96MHz/48MHz dependent on the blocks and Turbo Clock Control register

4.5.3 SYSTEM CLOCK

The SYSTEM CLOCK referred to as MCLK widely in this document is sourced from the 96MHz PLL or Dumb Ring Oscillator.

The MCLK clock domain is primarily driven by a 96MHz PLL, which derives 96MHz from the 32KHz clock domain. In Heavy Sleep mode, the 96MHz 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 MCLK clock domain until the PLL is stable. The PLL becomes stable after about 3ms; until that time, the 96MHz clock domain may range from 24MHz to 92MHz, as this is the accuracy range of the 60MHz ring. The 48MHz clock is derived from the 96MHz clock.MCLK can be configured to be 96MHz or 48MHz by setting Fast mode enable bit of Turbo Clock Control register

For achieving high performance the processor and PMC will run at 96MHz. The selection of 48MHz or 96Mhz is done by configuring the Fast mode enable in the Turbo Clock Control register. Only the below mentioned blocks clock are controlled by the Turbo Clock Control register. All other blocks operate with 48MHz clock under normal S0 State.

- ARM
- PMC
- Memory
- QMSPI
- Crypto Blocks

All other blocks will be running using 48MHz clock.

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 SOURCE

The 32kHz Clock Domain may be sourced by a crystal oscillator, using an external crystal, by an 32kHz Internal oscillator, or from a single-ended clock input. T he external single-ended clock source can itself be sourced either from the 32KHZ_IN signal that is a GPIO alternate function or from the XTAL2 crystal pin. The VTR source 32kHz Clock Register is used to configure the source for the 32 kHz clock domain.This clock source is used to drive the 96MHz PLL. Figure below represents the above information pictorially.

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.

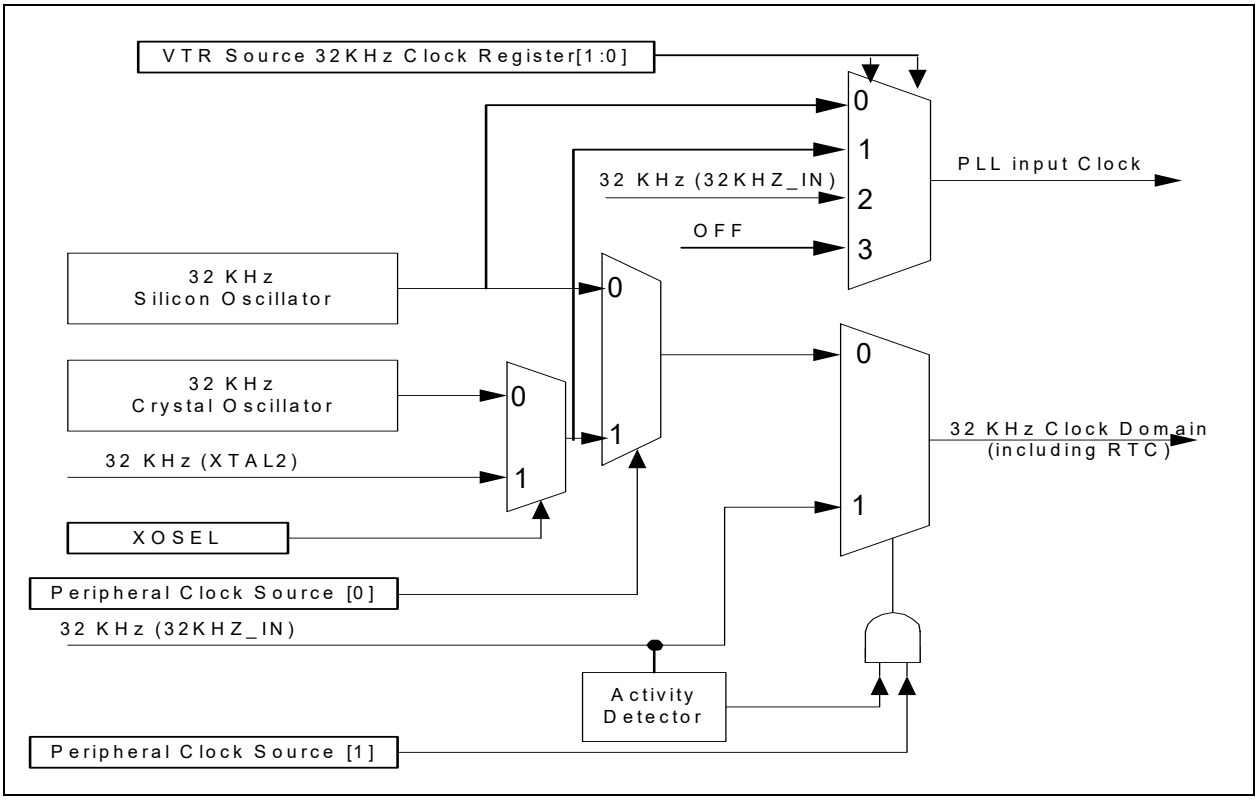
An external single-ended clock input for 32KHZ_IN may be supplied by any accurate 32KHz clock source in the system. The SUSCLK output from the chipset may be used as the 32KHz source. SUSCLK must be present when VTR is on. See chipset documentation for details on the use of SUSCLK.

If firmware switches the 32KHz clock source, the 96MHz PLL will be shut off and then restarted. The 96MHz clock domain will become unlocked and be sourced from the 60 MHz Ring Oscillator until the 96 MHz is on and locked.

4.5.5 32KHZ CORE CLOCK SOURCE

The 32KHz Core Internal Clock Source can be driven either by the 32.768 kHz Silicon Oscillator or the 32.768 kHz Crystal Oscillator. The VBAT SOURCE 32kHZ Register is used to configure the source for this 32 kHz clock domain. This clock is used by internal blocks that requires an always on low speed clock.

© 2021 Microchip Technology Inc.



4.5.6 32KHZ CRYSTAL OSCILLATOR

An External Crystal Oscillator can be used with EEC1727 for sourcing the 32kHz clock domain. For dual ended configuration, XTAL is connected between XTAL1 and XTAL2. Please refer EEC1727 PCB layout guide for details.

For Single ended XTAL configuration, external clock should be connected to XTAL2 pin and XTAL1 pin should be grounded. If the 32KHz source will never be the crystal oscillator, then the XTAL1 and XTAL2 pins should be grounded.

4.5.6.1 32KHz Crystal Oscillator Monitoring

This feature is optional and may be implemented in Application code. At power on the source for the 32kHz and 32kHz core will be the 32Khz Internal Oscillator. FW will monitor the external Crystal clock and may decide to switch the source of the 32kHz domain, if required, to the Crystal clock.

After a power on reset, the System clock source would run out of the 60 MHz Ring Oscillator until the PLL is locked. The source clock for the PLL should be selected by configuring the VTR source 32kHz Clock Register to 32kHz Internal Silicon Oscillator. Using the 48MHz PLL clock locked to the internal Silicon Oscillator, measure the Crystal clock frequency and after N good pulses are detected in a row, clock monitor asserts interrupt to the EC and the status register gives the interrupt status. At this point FW can change the source to Crystal as the source for the PLL reference clock.

- On VBAT POR, everything is disabled.Note that VBAT is connected to VTR1. See Note 4.
  - System Clock is 60 MHz Ring Oscillator; all 32kHz Clock sources are OFF
- Boot ROM enables the Internal silicon oscillators 32kHz Clock and sets it as the PLL Reference
  - Once PLL is locked, System Clock is driven by the PLL; 32kHz PLL reference clock is from Internal Silicon Oscillator
- Application firmware enables XTAL
- Application firmware sets up Clock Monitor Counter limits and IRQ’s
- Application firmware sets Time-out counter running in the background in case the clock is not within range.
- Application firmware enables XTAL Monitor Counter and XTAL Valid Counter
- Application firmware polls or waits for interrupt for XTAL to PASS or FAIL

- Application firmware switches PLL clock source to 32kHz XTAL clock, if it is Good.
  - Once PLL locks, System Clock is driven by PLL;32kHz PLL reference clock is from XTAL
- Application firmware disables all Monitor Counters to save power.

4.6 Resets

TABLE 4-8: 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.
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 M4F SYSRESETREQ
RESET_VCC	Performs a reset when Host power (VCC) is turned off	This signal is asserted if • RESET_SYS is asserted **Note:** The PWR_INV bit in the Power Reset Control Register is ‘1b’
RESET_HOST	Performs a reset when the system host resets the Host Interface.	This signal is asserted if • RESET_SYS is asserted • The PWR_INV bit in the Power Reset Control Register is ‘1b’
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

© 2021 Microchip Technology Inc.

TABLE 4-8: DEFINITION OF RESET SIGNALS (CONTINUED)

Reset	Description	Source
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.

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 4-9: 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.

TABLE 4-9: POWER MANAGEMENT PROTOCOL (CONTINUED)

Power State	SLEEP_ENABLE	CLOCK_REQUIRED	Description
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 96 MHz 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 4-11, "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 96 MHz 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 96 MHz 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 96 MHz clock domain is gated off, but the 96 MHz remains operational and locked to the 32KHz Core clock domain. On wake, the PLL output is ungated and the 96 MHz 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 96 MHz is shut down. On wake, the 60 MHz Ring Oscillator is used to provide a clock source for the 96 MHz clock domain until the PLL locks to the 32KHz Core 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 96 MHz clock will not operate correctly until the PLL locks.The time to lock latency for the PLL is shown in Table 4-11, "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.

4.7.4.1 Wake-Only Events

Some devices which respond to an external master require the 96 MHz clock domain to operate but do not necessarily require and immediate processing by the EC. Wake-only events provide the means to start the 96 MHz clock domain without triggering an EC interrupt service routine. This events are grouped into a single GIRQ, GIRQ22. Events that are

© 2021 Microchip Technology Inc.

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 4-10: 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	Turbo Clock Control
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
88h	Reserved
8Ch	VTR source 32kHz Clock Register
C0h	32kHz Period count Register
C4h	32kHz High pulse count Register
C8h	32kHz Period MIN count Register
CCh	32kHz Period MAX count Register
D0h	32kHz Duty Cycle variation Register
D4h	32kHz Duty Cycle variation Max Register
D8h	32kHz Valid Count Register
DCh	32kHz Valid Count MIN Register
E0h	32kHz Control Register
E4h	32kHz Source Interrupt Register
E8h	32kHz Source Interrupt ENABLE 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 Table 3-2, "Sleep Allocation".

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

© 2021 Microchip Technology Inc.

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 96 MHz clock domain to lock to 96 MHz. The wake latency in all sleep modes is 0ms. Table 4-11 shows the time to lock latency for the different sleep modes. 1=Heavy Sleep 0=Light Sleep	R/W	0h	RESET_SYS

TABLE 4-11: 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 96 MHz clock by 48(2MHz processor clock) 16=divide the 96 MHz clock by 16 (6MHz processor clock) 4=divide the 96 MHz clock by 4 (24MHz processor clock) 2=divide the 96 MHz clock by 2(48MHz processor clock) 1=divide the 96 MHz clock by 1 (96MHz 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

© 2021 Microchip Technology Inc.

4.9.8 PCR POWER RESET STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:11	Reserved	RES	-	-
10	32K_ACTIVE 1=The 32K clock input is present. 0=The 32K clock input is not present.	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_TRST# 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
3	RESET_HOST_STATUS Indicates the status of RESET_VCC. 1=Reset not active 0=Reset active	R	-	Note 1
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**
2	VCC_PWRGD_STATUS Indicates the status of VCC_PWRGD. 1=VCC_PWRGD asserted 0=VCC_PWRGD not asserted	R	xh	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**
31:9	Reserved	RES	-	-
8	Fixed to Logic 1. This bit must not be programmed to any other value.	R/W	1h	RESET_SYS
7:1	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. This bit is used by firmware to control the internal RESET_VCC signal function and the external PWROK pin. This bit is read-only when VCC_PWRGD is de-asserted low. The internal RESET_VCC signal is asserted when this bit is asserted even if the PWROK pin is configured as an alternate function.	R / R/W	1h	RESET_SYS

© 2021 Microchip Technology Inc.

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 TURBO CLOCK CONTROL

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2	Fast mode enable 0=48MHz Clock Operation 1=96MHz Clock Operation Only clock to QMSPI, ARM Processor, Memory, Crypto and PMC blocks are changed by this bit. All other peripherals run off the 48MHz clock.	R/W	0b	RESET_SYS
1:0	Reserved	RES	-	-

4.9.12 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

4.9.13 VTR SOURCE 32KHZ CLOCK REGISTER

Offset	8Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	RESERVED	-	-	-
1:0	PLL Reference Source 0=Internal Oscillator 1=XTAL 2=32kHz_IN VTR Pin 3=None (OFF) If set to 0x3, the PLL will not receive a reference clock and will be held in Reset	R/W	3h	RESET_SYS

4.9.14 32KHZ PERIOD COUNT REGISTER

Offset	C0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	32kHz Period Counter Counts System clock between 2 positive edges of a 32kHz Clock	RO	-	RESET_SYS

4.9.15 32KHZ HIGH PULSE COUNT REGISTER

Offset	C4h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	32kHz High Counter Counts how many System clock cycles the 32kHz clock remains High	RO	-	RESET_SYS

4.9.16 32KHZ PERIOD MIN COUNT REGISTER

Offset	C8h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	32kHz Period Minimum Counter This is the minimum period count that is acceptable for the 32kHz counter to flag a PASS status	R/W	0h	RESET_SYS

© 2021 Microchip Technology Inc.

4.9.17 32KHZ PERIOD MAX COUNT REGISTER

Offset	CCh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	32kHz Period Maximum Counter This is the maximum period count that is acceptable for the 32kHz counter to flag a PASS status	R/W	0h	RESET_SYS

4.9.18 32KHZ DUTY CYCLE VARIATION REGISTER

Offset	CCh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	32kHz Duty Variation Counter This is the difference in system clocks between the 32kHz clocks High Pulse Width and its Low Pulse Width.	RO	-	RESET_SYS

4.9.19 32KHZ DUTY CYCLE VARIATION MAX REGISTER

Offset	D4h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	32kHz Duty Variation Maximum This is the maximum variation allowed to generate a PASS condition for the 32kHz clock.	R/W	0h	RESET_SYS

4.9.20 32KHZ VALID COUNT REGISTER

Offset	D8h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	32kHz Valid Count This counts the number of valid 32kHz periods and pulse width variations measured in a row. This count increments on a PASS and will reset on a FAIL.	RO	-	RESET_SYS

4.9.21 32KHZ VALID COUNT MIN REGISTER

Offset	DCh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	32kHz Valid Count Minimum This is the minimum value of Counter 32kHz Valid Count that will flag the status Counter Valid.	R/W	0h	RESET_SYS

4.9.22 32KHZ CONTROL REGISTER

Offset	E0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	RESERVED	-	-	-
24	32kHz Clear Counters Clears the Counters	WO	0h	RESET_SYS
23:5	RESERVED	-	-	-
4	32kHz Source Selects the 32kHz Clock source thats is to be measured 0=XTAL 1=Internal Oscillator	R/W	0h	RESET_SYS
3	RESERVED	-	-	-
2	32kHz Valid Enable Enables the 32kHz valid counter	R/W	0h	RESET_SYS
1	32kHz Duty Cycle Counter Enable Enables the Duty Counter and checks for the 32kHz off of the system clock.	R/W	0h	RESET_SYS
0	32kHz Period Counter Enable Enables the Period Counter and checks for the 32kHz off of the system clock.	R/W	0h	RESET_SYS

© 2021 Microchip Technology Inc.

4.9.23 32KHZ SOURCE INTERRUPT REGISTER

Offset	E4h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	RESERVED	-	-	-
6	32kHz Unwell Interrupt This interrupt is set if there is any type of failure on the counters while monitoring the 32kHz clock (period or duty variation) after the Counter 32kHz Valid has been set. This interrupt is disabled if the counters are disabled or cleared, and will only be re-enabled after the next Counter 32kHz Valid is set.	R/W1C	-	RESET_SYS
5	32kHz Valid Interrupt This interrupt is set after the Valid Count check passes.	R/W1C	-	RESET_SYS
4	32kHz Stall Interrupt This interrupt is set when the 32KHz clock period counter over-flows.	R/W1C	-	RESET_SYS
3	32kHz Fail Interrupt This interrupt is set when either the period or duty variation checks fail on a 32kHz clock positive edge.	R/W1C	-	RESET_SYS
2	32kHz Pass Duty Interrupt This interrupt is set when the Duty Cycle Variation check passes on every 32kHz positive edge.	R/W1C	-	RESET_SYS
1	32kHz Pass Period Interrupt This interrupt is set when the period check passes on every 32kHz positive clock edge (passes Max/Min Period Check).	R/W1C	-	RESET_SYS
0	32kHz Pulse Ready Interrupt This interrupt is set on every positive edge of an 32kHz clock (after the 1st). This interrupt indicates that the status of the counters has been updated.	R/W1C	-	RESET_SYS

4.9.24 32KHZ SOURCE INTERRUPT ENABLE REGISTER

Offset	E8h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	RESERVED	-	-	-
6	32kHz Unwell Interrupt Enable	R/W	0h	RESET_SYS
5	32kHz Valid Interrupt Enable	R/W	0h	RESET_SYS
4	32kHz Stall Interrupt Enable	R/W	0h	RESET_SYS

Offset	E8h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	32kHz Fail Interrupt Enable	R/W	0h	RESET_SYS
2	32kHz Pass Duty Interrupt Enable	R/W	0h	RESET_SYS
1	32kHz Pass Period Interrupt Enable	R/W	0h	RESET_SYS
0	Counter 32kHz Pulse Ready Interrupt Enable	R/W	0h	RESET_SYS

© 2021 Microchip Technology Inc.

5.0 ARM M4F BASED EMBEDDED CONTROLLER

5.1 Introduction

This chapter contains a description of the ARM M4F Embedded Controller (EC).

The EC is built around an ARM® Cortex®-M4F Processor provided by Arm Ltd. (the “ARM M4F IP”). The ARM Cortex® M4F is a full-featured 32-bit embedded processor, implementing the ARMv7-M THUMB instruction set and FPU instruction set in hardware.

The ARM M4F 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 M4F is configured as follows.

- **Little-Endian** byte ordering is selected at all times
- **Bit Banding** is included for efficient bit-level access
- **Floating-Point Unit (FPU)** is included, to implement the Floating-Point instruction set in hardware
- **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
- **MPU** (Memory Protection Unit) is included for memory access controlSingle entry **Write Buffer** is incorporated

5.2 References

1. ARM Limited: Cortex®-M4F 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™-M4F 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-M4F

  The ARM designation for the specific IP selected for this product: a Cortex M4F processor core containing a hardware Floating Point Unit (FPU)

- 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®-M4F Technical Reference Manual, DDI0439C, 29 June 2010.

© 2021 Microchip Technology Inc.

- FPU

  Floating-Point Unit: a subblock included in the Core for implementing the Floating Point instruction set in hardware.

- 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.**

- MPU

  Memory Protection Unit.

- 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 EEC1727 EC. It manages accesses from both the EC processor and chip-level bus masters.

5.4 ARM M4F 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 EEC1727 IP has the following major external interfaces, as shown in Figure 5-1, "ARM M4F 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.

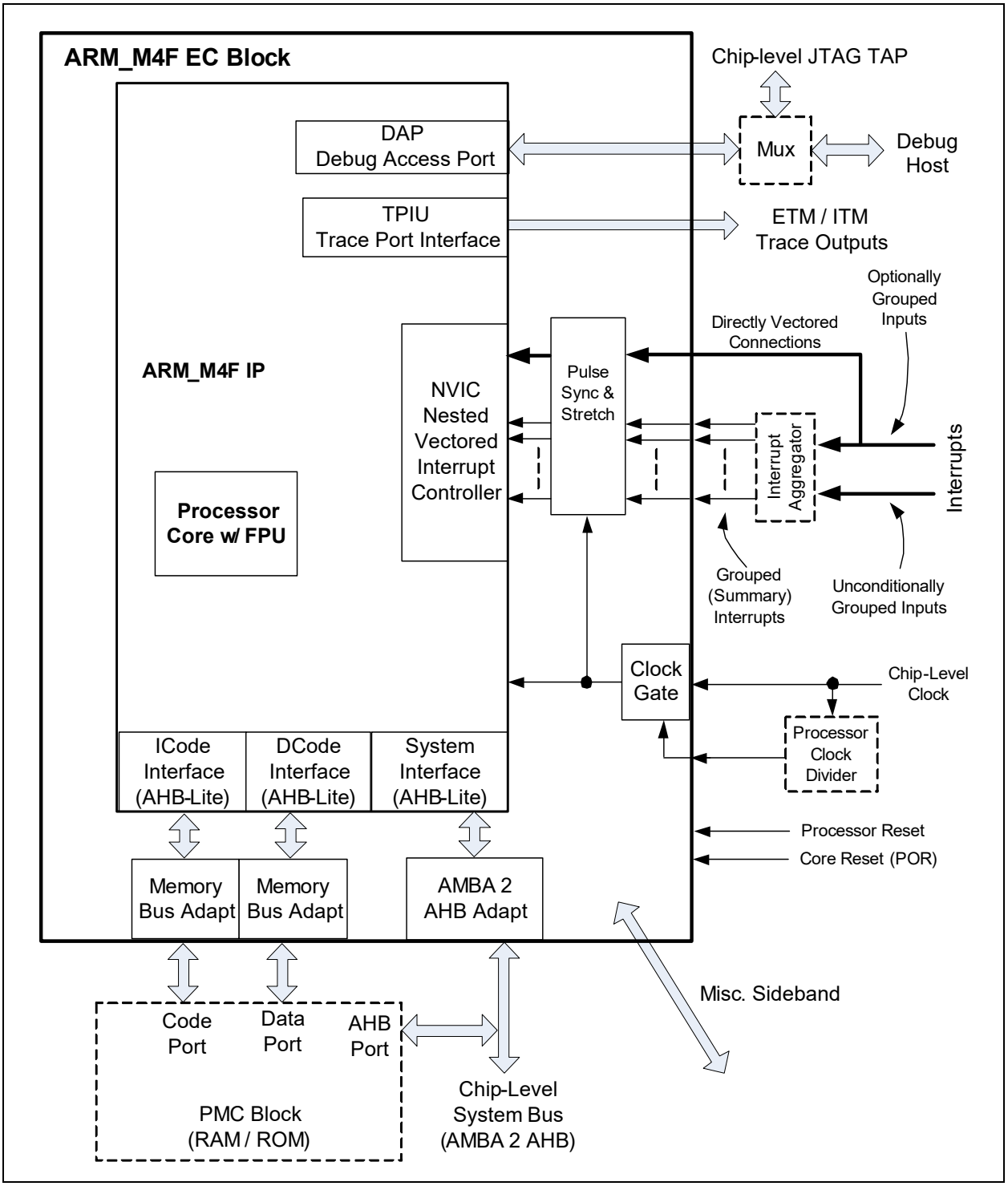
Aliased addressing spaces are provided at the chip level so that specific bus interfaces can be selected explicitly where needed. For example, the EC's Bit Banding feature uses the System AHB-Lite bus to access resources normally accessed via the DCode or ICode interface.

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.

© 2021 Microchip Technology Inc.

5.5 Block External Interfaces

FIGURE 5-1: ARM M4F 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 M4F 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
96 MHz	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 96 MHz 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 96 MHz 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 M4F Based Embedded Controller is reset by RESET_EC.

5.7 Interrupts

The ARM M4F 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 M4F 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.

© 2021 Microchip Technology Inc.

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 EEC1727 EC block, which are directly related to the IP bus ports. See Figure 5-1, "ARM M4F 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 M4F 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.

© 2021 Microchip Technology Inc.

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™-M4F Technical Reference Manual, DDI0440C, 29 June 2010.

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

© 2021 Microchip Technology Inc.

5.11 Delay Register

5.11.1 DELAY REGISTER

Offset	0800_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

© 2021 Microchip Technology Inc.

6.0 CACHE CONTROLLER

6.1 Introduction

The cache controller is a read only cache for the flash memory attached to the EC. Up to 32 MB of flash can be supported by the cache controller for cacheing.

- 32KB, 4-Way Set Associative Cache with 512B lines size
  - Each line is divided into 4 equal sections of 64 bytes

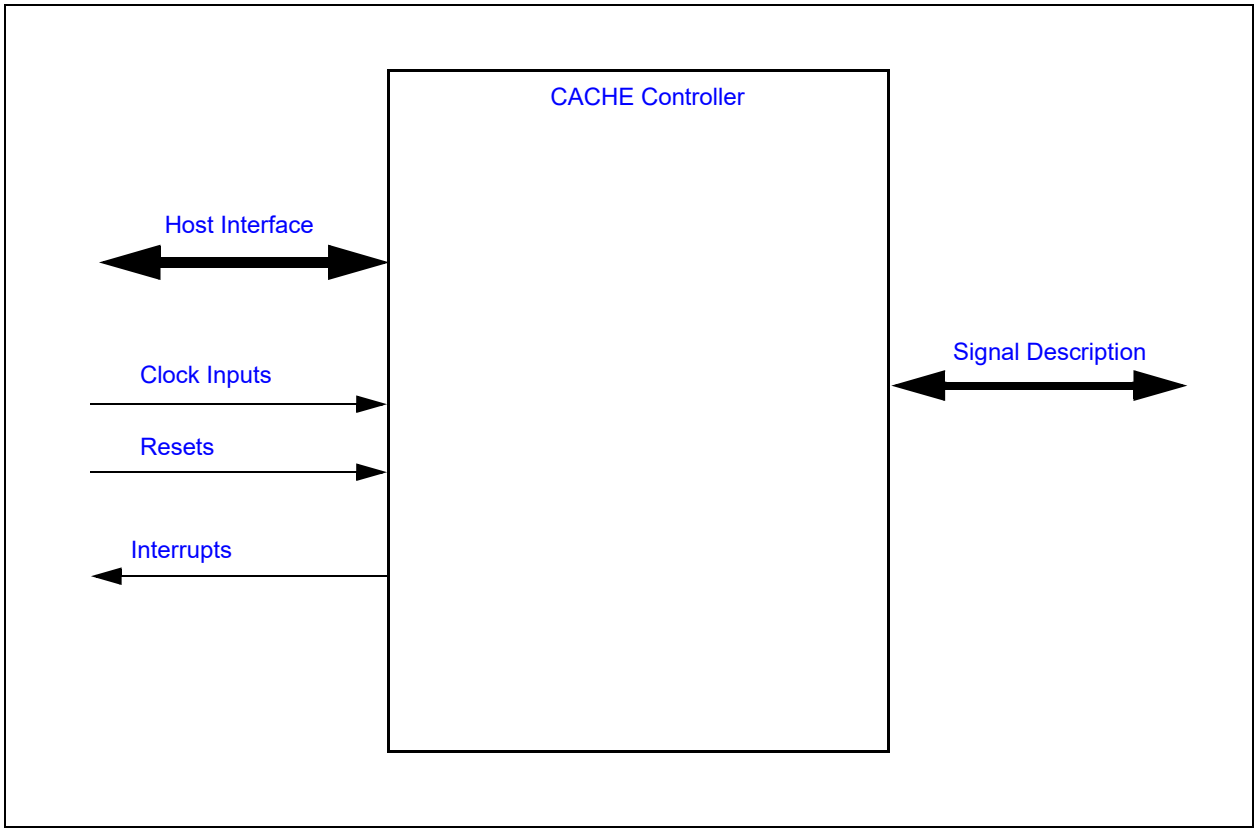
6.2 Terminology

- Section
  - Each 512Byte Line into smaller 64B sections for improving the miss latency.
- Critical Section First
  - The action where the cache will request data starting at the boundary of the target Section.
  - Data is returned to the processor when it is ready, instead of waiting for an entire cache line fill.
  - This reduces Miss Latency.
- Early Termination
  - The action where the cache will pre-terminate a cache line fill if another access requests data from the SPI.
  - Termination is on Section boundaries.
  - This reduces Miss Latency.
- Cache Replacement Policy
  - This is the algorithm used to determine which cache line is evicted when a Miss occurs and there are no empty cache lines to allocate.
  - Least Frequently Used (LFU) replaces the line which has been used least often.
  - Least Frequently Used with Dynamic Aging (LFUDA) is LFU, but also slowly decrements the hit rate count every time their is a miss-eviction.
  - Least Recently Used (LRU) replaces the line at the bottom of the stack. Lines get pushed onto the top of the stack whenever they are hit.
  - LRU with LFUDA (implemented algorithm) is a combination of LRU and LFUDA.
  - Cache implemented in the chip has LRU with LFUDA algorithm implemented
  - LFU count implemented is 2 and LRU count implemented is 4 i.e. 1 bit for LFU and 2 bits for LRU.
- Hit
  - The access resides in the cache.
- Hit/Fill
  - The access Region resides in the cache, but the data word has not been populated into the cache yet.
  - This can be due to Critical Word/Section First returning before filling the cache, or Early Termination aborting a full cache line fill.
- Miss
  - The access Region does not reside in the cache.
  - The cache will have to allocate a line for this access.

© 2021 Microchip Technology Inc.

6.3 Interface

FIGURE 6-1: I/O DIAGRAM OF BLOCK



6.4 Signal Description

There are no external signals for this block.

6.5 Host Interface

The registers defined for the CACHE Controller are accessible by the Embedded Controller as indicated in Section 6.10, "EC Registers".

6.6 Power, Clocks and Reset

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

6.6.1 POWER DOMAINS

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

© 2021 Microchip Technology Inc.

6.6.2 CLOCK INPUTS

Name	Description
96 MHz	This is the clock source to the Cache Controller

6.6.3 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.

6.7 Interrupts

There are no interrupts to EC from this block

6.8 Low Power Modes

The Cache Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. The SLEEP_EN and CLK_REG are attached to the APMC SLEEP_EN and CLK_REQ.

6.9 Description

The cache is a read only Cache. The Cache does not support writes of any sort.This Cache uses an 4-way Set Associative with a Least Recently Used (LRU) plus Least Frequently Used (LFU) with Dynamic Aging algorithm using Critical Section First and Early Termination. It has programmable fields for locking, invalidating and force filling.

The Cache consumes the lowest 32kB code SRAM with the start address of 0x000C_0000. When the Cache is in bypass mode (Cache SPI Enable = 0), this memory is accessed normally. If the Cache is active (Cache SPI Enable = 1), the memory is no longer usable as Code SRAM and is instead used as a Cache SRAM. If Activate bit is not set in the Cache Mode Register, this SRAM memory cannot be used by the processor as Code SRAM.

6.10 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 CACHE Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 6-1: REGISTER SUMMARY

Offset	Register Name
00	Cache Mode Register
04	Cache SPI Control
08	Cache SPI Data
0C	Cache SPI Bank
10	Cache Tag Validate
14	Cache Tag Validate Address
20	Cache Status
40	Cache Hit Hi
44	Cache Hit Lo
50	Cache Miss Hi
54	Cache Miss Lo
60	Cache Fill Hi
64	Cache Fill Lo

TABLE 6-1: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
C0-FF	Cache RX Buffer (64 Bytes)
800-807	Cache Tag Lock
C00-C07	Cache Tag Valid
1000-11FF	Cache Tag Address {0..31}

6.10.1 CACHE MODE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5	Invalidate Cache (invalidate_cache) Writing this field will trigger Cache flush	W	0	RESET_SYS or Soft Reset
4	Full line read enable 0: Critical section, first and early termination enabled 1: Critical section, first and early termination disabled	R/W	0	RESET_SYS or Soft Reset
3	RESERVED	RES	0	RESET_SYS or Soft Reset
2	Cache SPI Enable 0: Standard register access 1: Cache takes over the register interface of QMSPI	R/W	0	RESET_SYS or Soft Reset
1	Soft Reset Soft reset the Cache Controller module. This is self clearing bit	WO	0	RESET_SYS or Soft Reset
0	Activate 0: Disable block 1: Enable block	R/W	1	RESET_SYS or Soft Reset

© 2021 Microchip Technology Inc.

6.10.2 CACHE SPI CONTROL

Only valid when Controlling the QMSPI directly.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	RES	-	-

6.10.3 CACHE SPI DATA

Only valid when Controlling the QMSPI directly.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	RES	-	-

6.10.4 CACHE SPI BANK

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CSPI Bank This is OR'ed into the access address from the processor to create the address sent over to the SPI Flash. If the SPI Flash is only 24-bits, then bits [31:24] of this register have no effect. The LSB of this register should be reserved or set to 0x00 for all bits that are already controlled by the access address. For example, if the access address has a 512kB range, then bits [18:0] in this register should be 0x00.	R/W	0	RESET_SYS or Soft Reset

6.10.5 CACHE TAG VALIDATE

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	Tag Force Invalid This will take the cache line selected by Cache Tag Force Address and Invalidate it. 0: No Action 1: 1.A Tag Line will be de-allocated	WO	0	RESET_SYS or Soft Reset
2	Tag Force Fill Will cause the cache to immediately fill this cache line. 0: Line will not be filled until it is Miss/Fill 1: Line will auto fill immediately Works only with Tag Force Valid	WO	0	RESET_SYS or Soft Reset
1	Tag Force Lock Will lock the cache line. 0: Line will not be locked 1: Line will be locked Works only with Tag Force Valid	WO	0	RESET_SYS or Soft Reset
0	Tag Force Valid This bit will allow the cache select a Tag line, and allocate the Cache Tag Validate Address to it 0: No action 1: A tag line will be allocated	WO	0	RESET_SYS or Soft Reset

6.10.6 CACHE TAG VALIDATE ADDRESS

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Tag Force Address This is the address that will be stored in the Tag Line and accessed over SPI if a Tag Force Valid/Cache Tag Validate/Tag Force Invalid is issued. This address is still used in conjunction with the Cache SPI Bank.This is meant to be an address from the processors perspective. 4 Byte Boundary aligned	R/W	0	RESET_SYS or Soft Reset

© 2021 Microchip Technology Inc.

6.10.7 CACHE STATUS

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
2	SPI Error This flags when an error is detected while attempting to retrieve data from the SPI Flash.	R/W1C	0	RESET_SYS or Soft Reset
1	Invalidate Done This flags when the invalidate command has completed. The validate command is issued when the Tag Force Invalid field is set.	R/W1C	0	RESET_SYS or Soft Reset
0	Validate Done This flags when the validate command has completed. The validate command is issued when the Tag Force Valid field is set.	R/W1C	0	RESET_SYS or Soft Reset

6.10.8 CACHE HIT HI

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Hit count Hi Stores the Hit Count [63:32] of the Cache	R/W	0	RESET_SYS or Soft Reset

6.10.9 CACHE HIT LO

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Hit count Lo Stores the Hit Count [31:0] of the Cache	R/W	0	RESET_SYS or Soft Reset

6.10.10 CACHE MISS HI

Offset	50h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Miss count Hi Stores the Miss Count [63:32] of the Cache	R/W	0	RESET_SYS or Soft Reset

6.10.11 CACHE MISS LO

Offset	54h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Miss count Lo Stores the Miss Count [31:0] of the Cache	R/W	0	RESET_SYS or Soft Reset

6.10.12 CACHE FILL HI

Offset	60h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Fill Count Hi This is the Hit/Fill Count of the cache. This stores Count [63:32].	R/W	0	RESET_SYS or Soft Reset

© 2021 Microchip Technology Inc.

6.10.13 CACHE FILL LO

Offset	64h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Fill Count Lo This is the Hit/Fill Count of the cache. This stores Count [31:0].	R/W	0	RESET_SYS or Soft Reset

6.10.14 CACHE RX BUFFER (64 BYTES)

Offset	C0h-FFh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Rx Buffer This is for use with the SAF EC only. When the cache is requesting data from the SAF EC, this register will be set as the target address in the SAF. This will allow the cache to monitor when this buffer is written, so that it can take that data and place it into the cache.This will always be a 64 Byte region of memory that only exists to monitor bus traffic for the cache FSM.	WO	0	RESET_SYS or Soft Reset

6.10.15 CACHE TAG LOCK

Offset	800h-807h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
63:0	Tag Lock Setting this will lock a Tag Line so that the Cache Replacement Policy cannot evict this line. In the case where all Lines are locked, the Cache Replacement Policy will ignore the locks and evict a random Line. Do not lock all Lines in a Set. 0: Tag is not Locked 1: Tag is Locked	R/W	0	RESET_SYS or Soft Reset

© 2021 Microchip Technology Inc.

6.10.16 CACHE TAG VALID

Offset	C00h-C07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
63:0	Tag Valid Indicates Line has valid data in the cache. 0: Line is empty 1: Line is full	RO	0	RESET_SYS or Soft Reset

6.10.17 CACHE TAG ADDRESS {0..31}

Offset	1000h - 1080h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Tag Address This field stores the address that this Tag Line is accessing in the SPI Flash. This only stores the portion of the address that is important, meaning it does not store the LSB that corresponds to the bytes stored in the cache line, or the MSB that corresponds to the SPI Bank.	RO	Default value as per line the set belongs to	RESET_SYS or Soft Reset

© 2021 Microchip Technology Inc.

7.0 RAM AND ROM

7.1 SRAM

The EEC1727 contains two blocks of SRAM. 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.

Depending on the device, the first SRAM, which is optimized for code access, is

The second SRAM, which is optimized for data access, is

7.2 ROM

The EEC1727 contains a 128KB 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.

7.3 Additional Memory Regions

7.3.1 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.

7.3.2 RAM BIT-BAND REGION

The RAM bit-band region is an alias of the SRAM located at 2200_0000h, 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.

7.3.3 CRYPTOGRAPHIC RAM

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

7.3.4 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.

7.4 Memory Map

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

8.0 INTERNAL DMA CONTROLLER

8.1 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.

8.2 References

No references have been cited for this chapter.

8.3 Terminology

TABLE 8-1: TERMINOLOGY

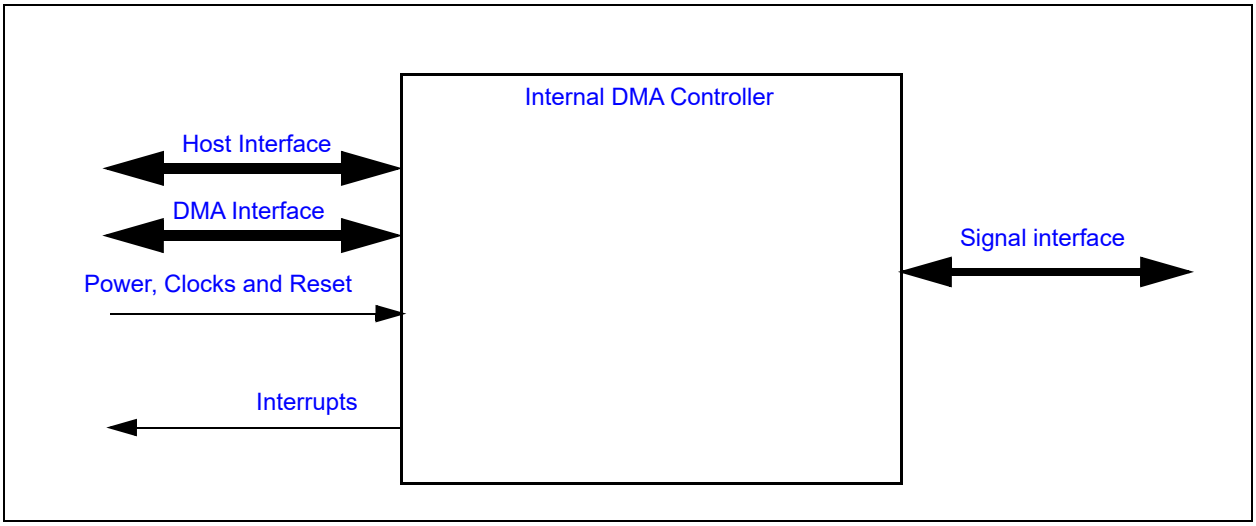
Term	Definition
DMA Transfer	This is a complete **DMA Transfer** which is done after the **Master Device** terminates the transfer, the Firmware Aborts the transfer or the DMA reaches its transfer limit. A DMA Transfer may consist of one or more data packets.
Data Packet	Each data packet may be composed 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.

© 2021 Microchip Technology Inc.

8.4 Interface

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

FIGURE 8-1: INTERNAL DMA CONTROLLER I/O DIAGRAM



8.5 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.

8.6 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".

8.7 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 EEC1727.

TABLE 8-2: 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 8-2: DMA CONTROLLER DEVICE SELECTION (CONTINUED)

Device Name	Device Number (Note 1)	Controller Source
QMSPI Controller	10	Transmit
	11	Receive
GP-SPI0 Controller	12	Transmit
	13	Receive
GP-SPI1 Controller	14	Transmit
	15	Receive
Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.		

TABLE 8-3: 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.

© 2021 Microchip Technology Inc.

TABLE 8-3: DMA CONTROLLER MASTER DEVICES SIGNAL LIST (CONTINUED)

Device Name	Dev Num	Device Signal Name	Direction	Description
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.
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.

TABLE 8-3: DMA CONTROLLER MASTER DEVICES SIGNAL LIST (CONTINUED)

Device Name	Dev Num	Device Signal Name	Direction	Description
GP-SPI0 Controller	12	SPI_TDMA_Req	INPUT	DMA request control from Quad SPI TX channel.
	13	SPI_RDMA_Req	INPUT	DMA request control from Quad SPI RX channel.
GP-SPI1 Controller	14	SPI_TDMA_Req	INPUT	DMA request control from Quad SPI TX channel.
	15	SPI_RDMA_Req	INPUT	DMA request control from Quad SPI RX channel.

8.8 Power, Clocks and Reset

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

8.8.1 POWER DOMAINS

TABLE 8-4: POWER SOURCES

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

8.8.2 CLOCK INPUTS

TABLE 8-5: CLOCK INPUTS

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

8.8.3 RESETS

TABLE 8-6: 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.

8.9 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 8-7: INTERRUPTS

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

8.10 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.

© 2021 Microchip Technology Inc.

8.11 Description

The EEC1727 features a 16 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 16 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 8.7, "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).

8.11.1 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.

8.11.2 OPERATION

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

8.11.2.1 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.

8.11.2.2 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.

8.11.2.3 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.

8.11.2.4 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 8-10, "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

8.11.2.5 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)

8.12 EC Registers

The DMA Controller consists of a Main Block and a number of Channels. Table 8-9, "Main Register Summary" lists the registers in the Main Block and Table 8-10, "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 8-8, "DMA Channel Offsets" to the offsets listed in Table 8-9, "Main Register Summary" or Table 8-10, "Channel Register Summary".

© 2021 Microchip Technology Inc.

TABLE 8-8: 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
DMA Controller	10	2C0h
DMA Controller	11	300h
DMA Controller	12	340h
DMA Controller	13	380h
DMA Controller	14	3C0h
DMA Controller	15	400h

TABLE 8-9: MAIN REGISTER SUMMARY

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

8.12.1 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

8.12.2 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 8-10: 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
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.

8.12.3 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

© 2021 Microchip Technology Inc.

8.12.4 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

8.12.5 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

8.12.6 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

8.12.7 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

© 2021 Microchip Technology Inc.

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

8.12.8 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

8.12.9 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

© 2021 Microchip Technology Inc.

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

8.12.10 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

8.12.11 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

8.12.12 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

© 2021 Microchip Technology Inc.

8.12.13 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

8.12.14 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

8.12.15 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

9.0 CHIP CONFIGURATION

9.1 Introduction

This chapter defines the mechanism to configure the device. Each logical device or block in the design has their own set of configuration registers. 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 Terminology

This section documents terms used locally in this chapter. Common terminology that is used in the chip specification is captured in the Chip-Level Terminology section.

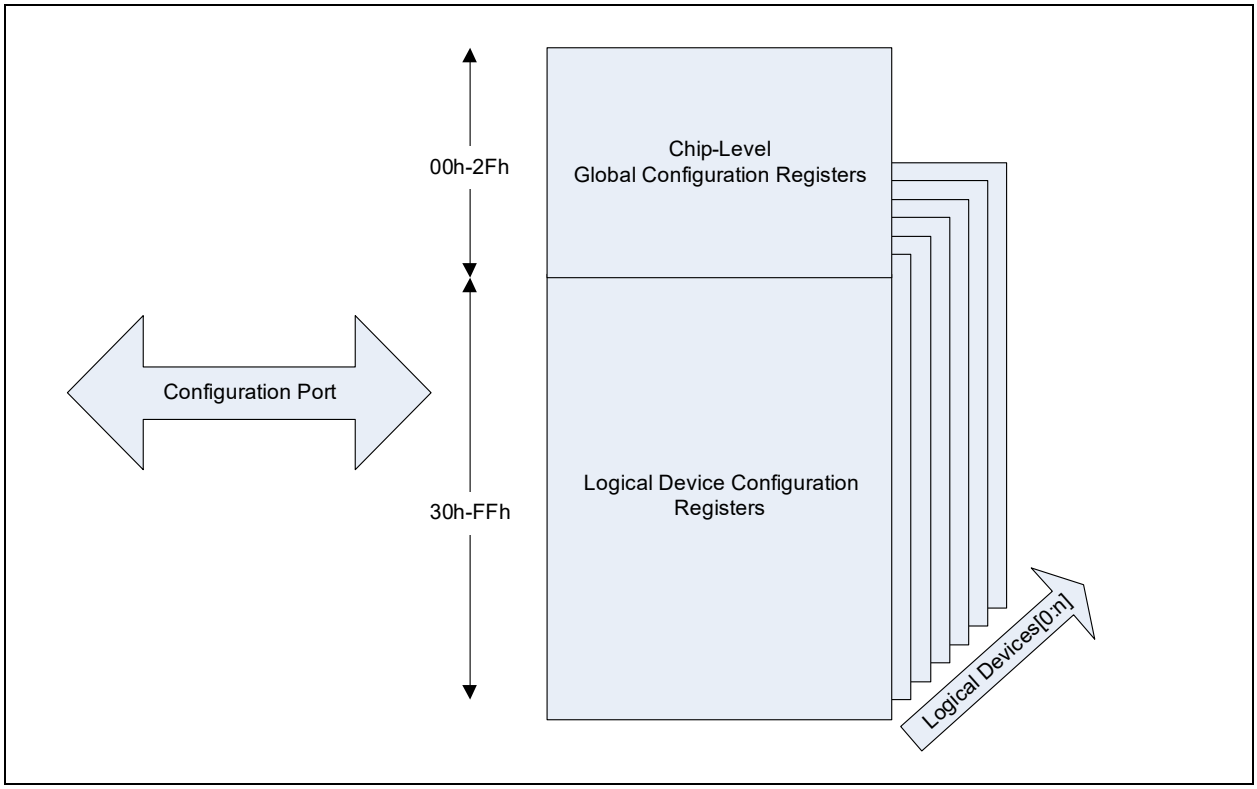
TABLE 9-1: TERMINOLOGY

Term	Definition
Global Configuration Registers	Registers used to configure the chip that are always accessible via the Configuration Port
Logical Device Configuration Registers	Registers used to configure a logical device in the chip. These registers are only accessible via the Configuration Port when enabled via the Global Configuration registers.

9.3 Interface

This block is designed to be accessed via the Host accessible Configuration Port.

FIGURE 9-1: BLOCK DIAGRAM OF CONFIGURATION PORT



Note: Each logical device has a bank of Configuration registers that are accessible at offsets 30h to FFh via the Configuration Port. The Logical Device number programmed in offset 07h determines which bank of configuration registers is currently accessible.

© 2021 Microchip Technology Inc.

9.4 Power, Clocks and Reset

This section defines the Power, Clock, and Reset input parameters to this block.

9.4.1 POWER DOMAINS

TABLE 9-2: POWER SOURCES

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

9.4.2 CLOCK INPUTS

This block does not require any special clock inputs.

9.4.3 RESETS

TABLE 9-3: RESET SIGNALS

Name	Description
RESET_SYS	Power on Reset to the entire device. This signal resets all the register and logic in this block to its default state.
RESET_HOST	A reset that occurs when VCC is turned off or when the system host resets the Host Interface.

9.5 Interrupts

This block does not generate any interrupts.

9.6 Low Power Modes

This block always automatically adjusts to operate in the lowest power mode.

9.7 Description

The Chip Configuration Registers are divided into two groups: Global Configuration Registers and Logical Device Configuration registers.

9.8 Configuration Registers

The EC can access Configuration Registers directly. 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.

All Global Configuration registers are accessible to the Host through the Configuration Port for all Logical Devices. at offsets 00h through 2Fh.

TABLE 9-4: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS

Register	Host Offset	Description
Chip (Global) Control Registers		
Reserved	00h - 01h	Reserved - Writes are ignored, reads return 0.
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.

TABLE 9-4: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS (CONTINUED)

Register	Host Offset	Description
Logical Device Number	07h	A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device. **Note:** The Activate command operates only on the selected logical device.
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.
OTP ID	24h	This register contains the OTP ID.
Validation ID	25h	This register contains the Validation ID
Boot ROM Revision ID	26-27h	This register contains the Boot ROM revision ID
TEST	28h - 2Fh	TEST. This register locations are reserved for Microchip use. Modifying these locations may cause unwanted results.

Note: Device Revision reports Current Revision and is Read-Only.

© 2021 Microchip Technology Inc.

10.0 EC INTERRUPT AGGREGATOR

10.1 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.

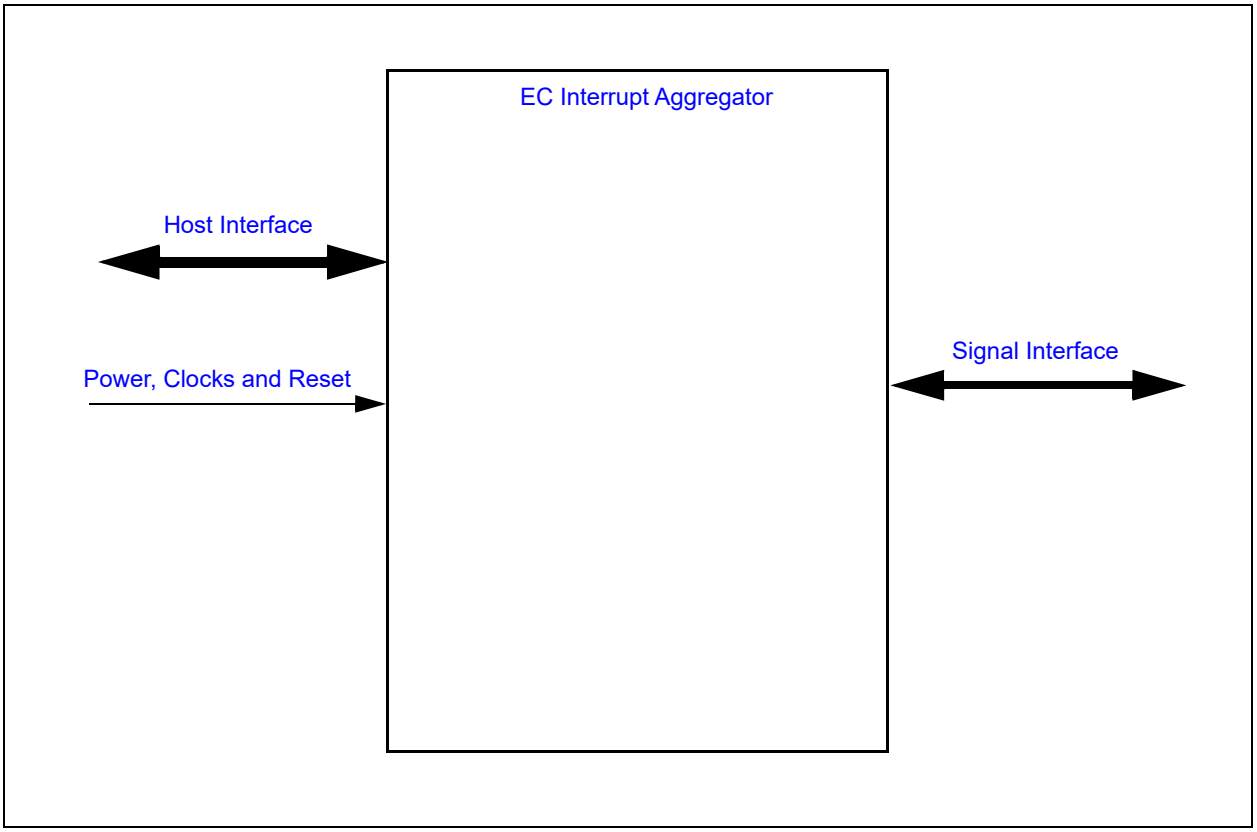
Interrupts classified as wake events can be recognized without a running clock, e.g., while the EEC1727 is in sleep state.

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

10.2 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 10-1: EC INTERRUPT AGGREGATOR INTERFACE DIAGRAM



10.3 Signal Description

10.3.1 SIGNAL INTERFACE

There are no external signals for this block.

10.4 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".

10.5 Power, Clocks and Reset

10.5.1 BLOCK POWER DOMAIN

TABLE 10-1: BLOCK POWER

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

10.5.2 BLOCK CLOCKS

TABLE 10-2: CLOCK INPUTS

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

10.5.3 BLOCK RESET

TABLE 10-3: 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.

10.6 Interrupts

This block aggregates all the interrupts targeted for the embedded controller into the Source Registers defined in Section 10.9, "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.

10.7 Low Power Modes

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

10.8 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.

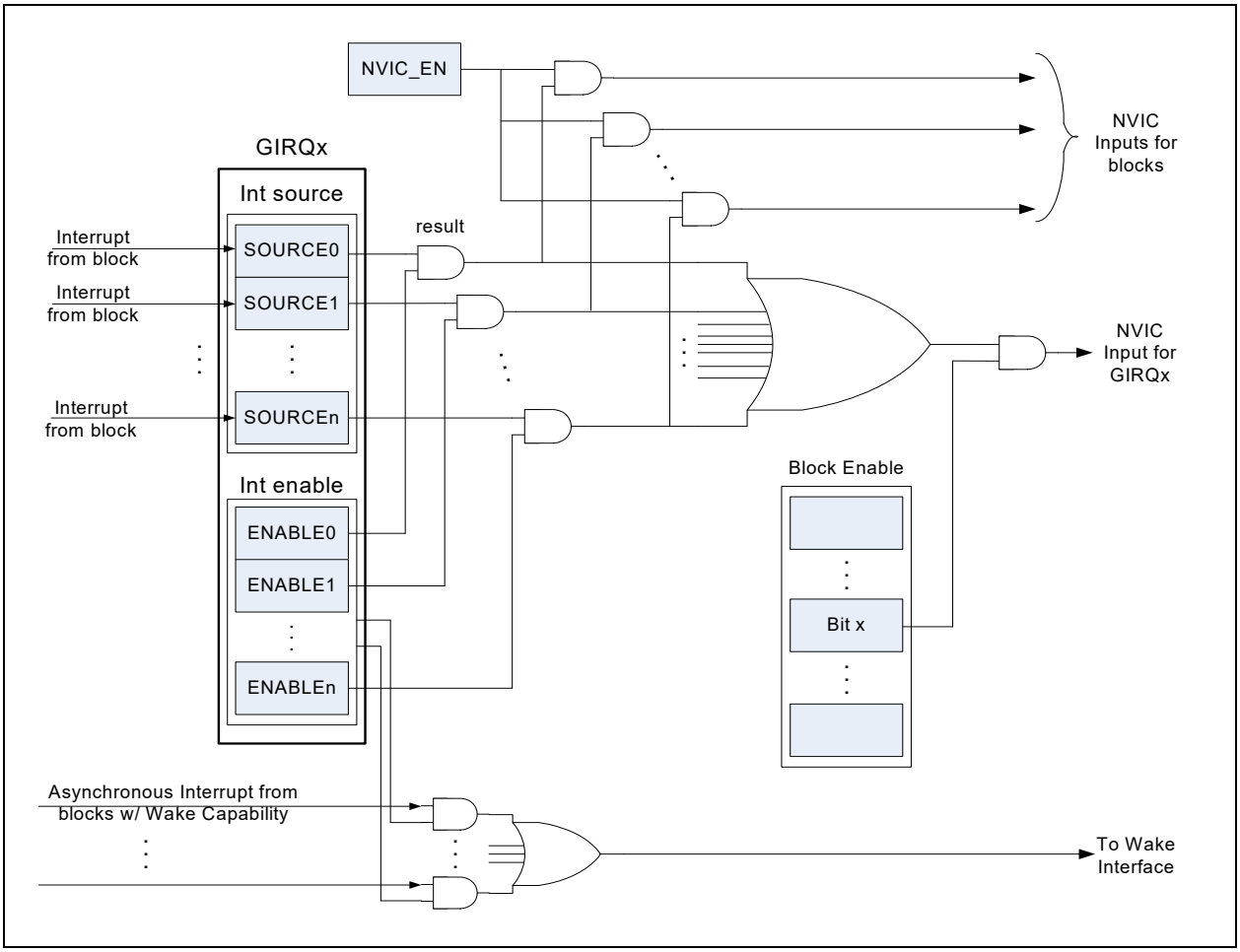
The Status and Enable are latched registers. There is one set of Enable register bits; both the Enable Set and Enable Clear registers return the same result when read. The Enable Set interface is used to set individual bits in the Enable register, and the Enable Clear is used to clear individual bits. 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 33.0, "EC Subsystem Registers"

Section 10.8.1 shows a representation of the interrupt structure.

© 2021 Microchip Technology Inc.

FIGURE 10-2: INTERRUPT STRUCTURE



10.8.1 AGGREGATED INTERRUPTS

All interrupts are routed to the ARM processor through the ARM Nested Vectored Interrupt Controller (NVIC). As shown in Figure 10-2, "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.

10.8.2 WAKE GENERATION

Wake-capable interrupts are listed in Table 3-3, "GPIO 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 60 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 60 MHz Ring Oscillator operation from a sleep state and wake the processor.

10.8.2.1 Wake Capable Interrupts

All GPIO inputs are wake-capable. In order for a GPIO input to wake the EEC1727 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.

10.8.2.2 Wake-Only Events

Some devices which respond to an external master require the 96 MHz clock domain to operate but do not necessarily require and immediate processing by the EC. Wake-only events provide the means to start the 96 MHz 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.

10.8.3 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.

10.8.4 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.

10.9 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 10-4: 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
28h	GIRQ10 Source Register
2Ch	GIRQ10 Enable Set Register

© 2021 Microchip Technology Inc.

TABLE 10-4: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
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
D8h	Reserved
DCh	GIRQ19 Source Register

TABLE 10-4: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
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 10.8.1, "Aggregated Interrupts".

© 2021 Microchip Technology Inc.

10.9.1 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

10.9.2 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

10.9.3 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

10.9.4 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

10.9.5 BLOCK ENABLE SET REGISTER

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

© 2021 Microchip Technology Inc.

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	-	-

10.9.6 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	-	-

10.9.7 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	-

11.0 SERIAL PERIPHERAL INTERFACE (SPI) SLAVE

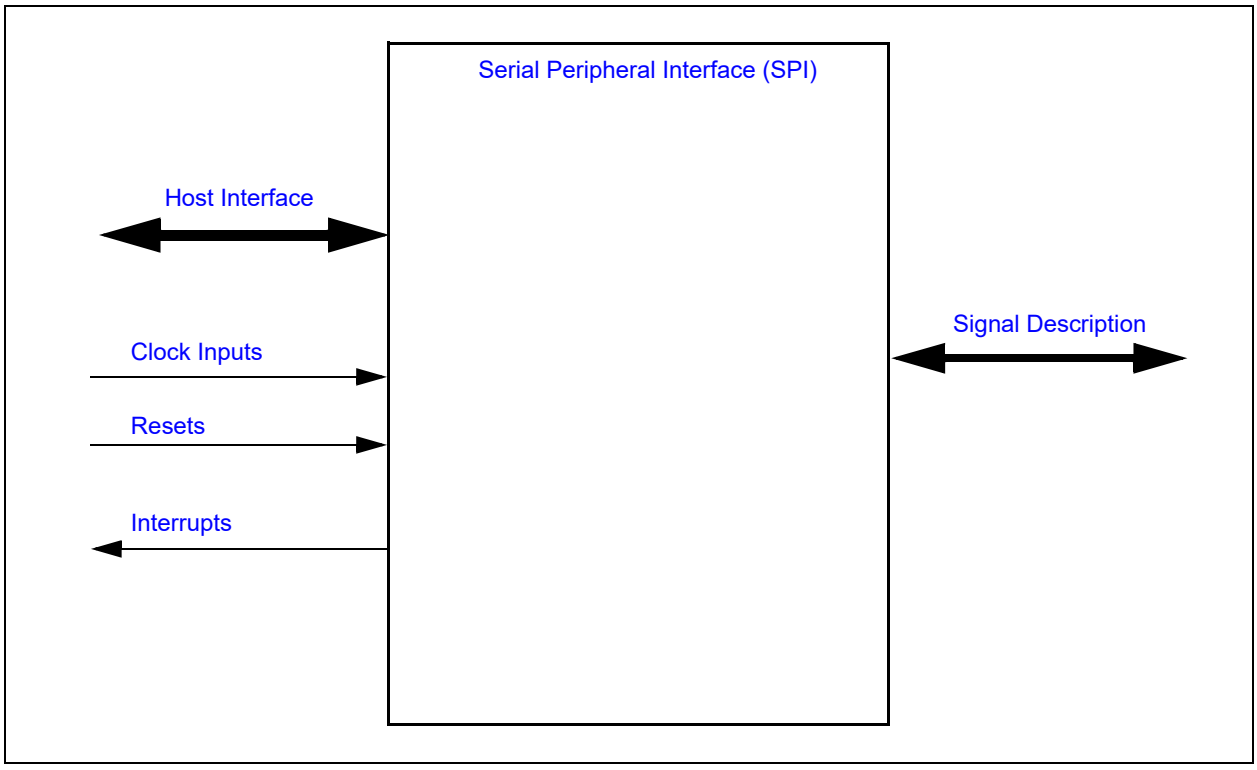
11.1 Introduction

The Serial Peripheral Interface (SPI) Slave provides a standard run-time mechanism for the SPI Master to communicate with the Embedded Controller (EC) and other logical components. The SPI includes 2 byte-addressable registers (16 bit SPI address field from SPI Master) in the SPI Master's address space, as well as by the EC. The SPI slave includes a DMA and once it is configured and enabled by the EC, can be used by the SPI Master to access bytes of memory designated by the EC without requiring any assistance. The SPI Slave provides a set of commands to access SPI Slave internal registers, designated SRAM memory/peripheral area within EC. In order to provide lower wait time for the SPI Master, the Serial Peripheral Interface (SPI) Slave provides posted Read/Write commands. In order to support posted and Non posted read/write, the SPI Slave has implemented FIFO, registers and mailbox in the block.

11.2 Interface

This block is designed to be accessed externally and internally via a register interface.

FIGURE 11-1: I/O DIAGRAM OF BLOCK



11.3 Host Interface

The registers defined for the Serial Peripheral Interface (SPI) Slave are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

11.4 Signal Description

The registers defined for the Serial Peripheral Interface (SPI) Slave are accessible by the SPI Master and the Embedded Controller (EC) as indicated in Section 11.9, "Configuration and Runtime Registers".

© 2021 Microchip Technology Inc.

TABLE 11-1: SPI SLAVE PORTS

Name	Direction	Description
SLV_SPI_SCLK	INPUT	Clock signal from SPI Master.
SLV_SPI_CS#	INPUT	Chip Select for SPI Slave from Master.
SLV_SPI_IOx	INOUT	SPI Slave data pins to Master. This is a 4 bit data bus.
SLV_SPI_MSTR_INT	OUTPUT	This is Hardware triggered interrupt is for the SPI Master and is asynchronous to SLV_SPI_SCLK.

11.5 Power, Clocks and Reset

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

11.5.1 POWER DOMAINS

The internal circuit of SPI Slave works on the VTR_CORE power domain as listed in Table 11-2 below. Please see Section 2.4.10, "Pin Multiplexing" to know the IO voltage supported by the ports listed in Table 11-1, "SPI Slave Ports".

TABLE 11-2: POWER SOURCES

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

11.5.2 CLOCK INPUTS

This block has two clock inputs as listed in Table 11-3, "Clock Signals". For both the 48MHz and SPI_CLK domain crossing low latency clock domain crossing synchronizers are used. Both clock are treated as asynchronous to each other.

TABLE 11-3: CLOCK SIGNALS

Name	Description
SLV_SPI_SCLK	This is the SPI clock from the Master. All SPI transfers take place with respect to this clock.
48MHz	Clock used for EC register access

11.5.3 RESETS

Resets to the SPI Slave are from the system reset which will reset the entire block or a write to the self-clearing reset bit. The SLV_SPI_CS# signal de-assertion is treated as reset to the SPI Interface state machines to take care of early termination of transfer by the SPI Master.

Note:	At SLV_SPI_CS# de-assertion, if there is any data still left in the RX FIFO SPI Slave will continue its process of emptying out, still generating AHB transfers. For TX FIFO, because there is pre-fetching as soon as the EC signals data is available, there will still be data left in the FIFO that SPI Master hasn't read out, so hardware will clear TX FIFO, clearing the FIFO of any contents and be available for the next transaction.

TABLE 11-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all the logic and register in this block.

11.6 Interrupts

This section lists the Interrupt pins from this block. Refer to Table 11-5 below for details.

TABLE 11-5: SYSTEM INTERRUPTS

Source	Description
SLV_SPI_MSTR_INT	This interrupt is for the SPI Master and is asynchronous to clock.

The SLV_SPI_MSTR_INT signal is asserted when any of the enabled interrupt in SPI Interrupt Enable Register is set and the corresponding condition for interrupt assertion is met.

TABLE 11-6: EC INTERRUPTS

Source	Description
SPI_EC_INTERRUPT	This interrupt is synchronous to the EC clock domain and is for the EC firmware.

The SPI_EC_INTERRUPT signal is asserted when any of the enabled interrupt in EC Interrupt Enable Register is set and the corresponding condition for interrupt assertion is met.

11.7 Low Power Modes

The Serial Peripheral Interface (SPI) Slave automatically enters low power mode when no transaction is targeted to it. The SPI Slave is a wake interface; at de-assert of chip select a wake event will occur.

11.8 Description

Some of the features of this block are listed below

1. SPI Slave module supports Simple Mode (SM) and Advanced Mode (AM).

Simple Mode

When the requirement is for a EC firmware controlled data flow, this mode will become helpful. No commands are supported in this mode and the data from the SPI Master is passed to the EC for interpreting and taking appropriate action on it. The flow control is implemented in EC firmware/software by the end user. There are no interrupts to EC (SPI_EC_INTERRUPT) and SPI Master (SLV_SPI_MSTR_INT) available in this mode.

1. SPI Slave module is Wake Capable.
2. SPI Slave module supports Single Wire and Mode 0 / Mode 3 transfers in this mode. SPI Communication Configuration Register configuration settings are ignored.
3. SPI Slave module only supports byte transfer with Undefined length in this mode.
4. SPI Slave module supports only one window with programmable Memory Base Address0 Register, Memory Write Limit0 Register for write data and Memory Base Address1 Register, Memory Read Limit1 Register for reads.
5. SPI Slave module Interrupt are don't care in Simple Mode.

Note:	Since there are no interrupts available in simple mode, the work around is to use GPIO pin interrupts. One could look for the GPIO on which SLV_SPI_CS# is present in from Section 2.3, "Pin List". Please refer to GPIO Pin control register bits [6:4] for setting up the interrupt for SLV_SPI_CS# when it is asserted. This way EC_FW can have a notification that SPI Master is about the transfer data to the EC.

6. SPI Master cannot directly access the SPI Slave registers listed in Table 11-8, "Register Summary" through commands listed in Table 11-9, "SPI Commands", in this mode. The SPI Master and EC Firmware will have to implement a protocol to make the EC Firmware read the SPI slave registers and send it to SPI Master if needed.
7. SPI Slave module supports Full Duplex mode.
8. This mode uses an application code (Software) controlled data flow. The SPI Slave blindly transfer the read/write data to the SRAM for EC Firmware to interpret. See Section 11.11.5, "Simple Mode" for transfer data format.
9. SPI Slave module uses byte counter to count the number of Bytes received or transmitted.
10. The Max packet length of an undefined length transfer is 32K Byte but recommended the master limits the size according to the limits placed by the EC. The data above the value written in Memory Write Limit0 Register will be ignored for writes and data above the value read from Memory Read Limit1 Register will be invalid data for reads.
11. The wake up timing of the SPI Slave have to be accounted for by the SPI Master.

- The SPI Master has to wait for the wake up timing requirements from light and heavy sleep after asserting SLV_SPI_CS# and before initiating the read / write transfer. This time is required for the clock to be available.

© 2021 Microchip Technology Inc.

- In case of light sleep, the clock is gated and will be available fairly quickly.
- However in the case of heavy sleep state of the chip, the PLL is off and PLL has to come up and lock for all blocks to be functioning properly.

Advanced Mode

1. SPI Slave module is Wake Capable.
2. SPI Slave module supports Single / Quad Wire and Mode 0 / Mode 3 transfers.
3. SPI Slave module supports programmable number of turn-around (TAR) cycles for Quad mode. Please see Section 11.9.1, "SPI Communication Configuration Register" bits [9:8].

Note: TAR Time are like dummy cycles and are used to introduce wait states. Writing some of the internal SPI Slave registers, SRAM Memory requires clock domain transfer and therefore the status of the write operation may not be available in the successive clock. For such transfers, turn around time will be used for signal direction to change and dummy cycles are used to account for clock domain transfer and completion of operation requested by the SPI Master.

4. SPI Slave module supports standalone 8,16, 32 bit transfers and block transfers of 2-8 DWords read/write memory accesses with error response. Please refer Table 11-9, "SPI Commands" for the list of commands SPI Master can issue and command format is in Section 11.10.1, "Command Format".
5. All the SPI Slave commands listed in Table 11-9, "SPI Commands" are processed by hardware within SPI Slave block, ensuring minimum possible latency
6. SPI Slave module supports Base Address Enable and Memory Access Window of 256 – 4K bytes and error if disabled or out of range.
7. SPI Slave module supports Poll command, described in Section 11.10.1.9, "Poll Command Format", for quick read of status register.
8. SPI Slave module supports Status Register which will not be transaction specific, if not cleared after ever transfer from the SPI Master. There are a set of flags for errors or done transactions for Master or System to be aware. Please refer to Section 11.9.2, "SPI Slave Status Register" and Section 11.9.3, "SPI EC Status Register" for details.

Note: There are two set of SPI Slave status and Interrupt enable registers provided in the design. One set is in the SPI clock domain and is directly accessible by the SPI Master using CMD_EXT_REG_W8 and CMD_EXT_REG_R8 commands. This Status register is described in Section 11.9.2, "SPI Slave Status Register" and interrupt enable register is described in Section 11.9.4, "SPI Interrupt Enable Register". The second set of SPI Slave status and Interrupt enable register is in the EC clock domain and is accessible to EC alone. This Status register is described in Section 11.9.3, "SPI EC Status Register" and interrupt enable register is described in Section 11.9.5, "EC Interrupt Enable Register".

9. SPI Slave module supports important set of commands (described in Section 11.10, "Commands Supported") to allow direct access to the SPI Slave's registers, which are in EC clock domain, using SREG commands (CMD_SREG_W8, CMD_SREG_W16, CMD_SREG_W32, CMD_SREG_R8, CMD_SREG_R16 and CMD_SREG_R32) with 8, 16, 32 bit size. SREG stands for SPI Slave Registers. All the registers listed in Table 11-8, "Register Summary", except Section 11.9.3, "SPI EC Status Register" and Section 11.9.5, "EC Interrupt Enable Register", are accessible to the SPI Master before they are locked.
10. SPI Slaves module allows configuration registers to be locked. Please see note under Section 11.9, "Configuration and Runtime Registers" for details.
11. SPI Slave module supports accesses to external register bank situated in the SLV_SPI_SCLK domain with 8-bit Read and Write Commands. This set of command is used when the register in the SPI slave is in the same clock domain as SPI Slave. These commands do not require wait cycles. Please see Section 11.10.1.10, "External Register Write Command Format" and Section 11.10.1.11, "External Register Read Command Format". The commands are CMD_EXT_REG_W8 and CMD_EXT_REG_R8. Registers that can be accessed are given below.

- SPI Slave Status Register
- SPI Interrupt Enable Register
- SPI Master-to-EC Mailbox Register
- EC-to-SPI Master Mailbox Register

12. SPI Slave module supports programmable Wait time for transactions between SLV_SPI_SCLK and 48MHz EC clock domain. Please refer Section 11.10.1.1, "Non-Posted Memory (Block) and SREG Write Command Format" and Section 11.10.1.2, "Non-Posted Memory (Block) and SREG Read Command Format". The commands that need this support are listed below

- Register Write commands CMD_SREG_W8, CMD_SREG_W16, CMD_SREG_W32.
- Register Read commands CMD_SREG_R8, CMD_SREG_R16, CMD_SREG_R32
- Non posted Memory Write commands CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32
- Non posted Memory Read commands CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32

Note:	The registers and memory accessed using these commands are implemented in EC clock domain, requiring wait cycles for signals and data to cross clock domain. During the wait time, the SPI slave will transfer lower 8bit of the SPI Slave Status Register which contain the current transfer status.

13. SPI Slave module supports programmable interrupt enables for both the EC firmware (SPI_EC_INTERRUPT) and the SPI Master (SLV_SPI_MSTR_INT).
14. SPI Slave module supports separate interrupt (SPI_EC_INTERRUPT) to the EC. EC can enable the various interrupts for which processor will get the interrupt by setting the appropriate bits of Section 11.9.5, "EC Interrupt Enable Register".
15. SPI Slave module supports separate interrupt (SLV_SPI_MSTR_INT) to the SPI Master. The SPI Master or EC can enable the various interrupts for which SPI Master will get the interrupt by setting the appropriate bits of Section 11.9.4, "SPI Interrupt Enable Register".

The Serial Peripheral Interface (SPI) Slave is composed of register interface, Memory interface and a mailbox interface. Fully on the SPI CLK domain, the SPI_IF's function is to transmit and receive data to and from the SPI Master using the SPI protocol. The block captures the incoming command and along with the dispatcher units determine if the rest of the command can be accepted.

11.8.1 SPI CLOCK FREQUENCY SUPPORTED

The Table 11-7, "Supported SLV_SPI_SCLK clock Frequency" lists the supported SPI clock frequency of this block. Running the chip outside the specified SLV_SPI_SCLK clock frequency may cause unspecified results.

TABLE 11-7: SUPPORTED SLV_SPI_SCLK CLOCK FREQUENCY

SPI Slave Mode	Supported SLV_SPI_SCLK Frequency
Single Mode	1MHz to 48 MHz
Advanced Mode Byte Command (Single Wire Interface)	1MHz to 48 MHz
Advanced Mode Byte Command (Quad Wire Interface)	1MHz to 48 MHz
Advanced Mode DWORD Command (Single Wire Interface)	1MHz to 48 MHz
Advanced Mode DWORD Command (Quad Wire Interface)	1MHz to 48 MHz

Note:	Byte command is for reading/writing 1 byte of data in one transfer.
Note:	DWORD command is for reading/writing 4 byte of data in one transfer.
Note:	Undefined length Read/Write is for data transfer. SPI Slave registers cannot be read/written by SPI Master using this command.
Note:	Using DWORD command for lengthy transfers is better as they utilize the internal bus bandwidth better.

11.8.2 EMBEDDED MEMORY MAP

Each Serial Peripheral Interface (SPI) Slave provides direct access for the SPI Master into two windows of 32K Byte each in the EC's internal address space. This mapping is programmable through a register, programmed during boot up:

© 2021 Microchip Technology Inc.

The Base addresses, the Read limits and the Write limits are defined by registers that are in the EC address space and cannot be written by the SPI Master if the register is locked. In each region, the Read limit need not be greater than the Write limit. The regions can be non-contiguous, contiguous or overlapping.

Each window into the EC memory can be as large as 32k bytes in the 32-bit internal address space. In Advanced Mode, the register Memory Base Address0 Register defines the address that SPI Master can write/read data to in EC space and register Memory Base Address1 Register defines the second set of address that the SPI Master can write/read data from the EC space.

In Simple Mode, the register Memory Base Address0 Register defines the address that SPI Master can write data to in EC space and register Memory Base Address1 Register defines the second set of address that the SPI Master can read data from the EC space.

11.8.3 EC AND SPI MASTER DATA REGISTERS

There are 16 32-bit EC registers as listed in Table 11-8, "Register Summary". The global lock register bit (Mask EC Registers) determines the type of access for SPI Master. Once the register is locked, the SPI Master can only read the data from these registers. Please see below note for the exceptions

Note: If Mask EC Registers bit is set in the System Configuration Register then only SPI Slave Status Register, SPI Master-to-EC Mailbox Register, EC-to-SPI Master Mailbox Register and SPI Interrupt Enable Register are accessible to the SPI Master.

11.9 Configuration and Runtime Registers

The registers listed in the below Table 11-8, "Register Summary" table are for a single instance of the Serial Peripheral Interface (SPI) Slave. EC access for each register listed in this table is defined as an offset to the SPI Slave module base address. The Base address of this block is listed in Table 3-1, "Base Address".

For EC firmware/software each register address is formed by adding the Base Address of the Serial Peripheral Interface (SPI) Slave instance to the offset address of the register as shown in the “Offset” column in Table 11-8, "Register Summary".

Note: From the SPI Master perspective, Access to SPI Slave requires SLV_SPI_CS# to be asserted followed by appropriate command/address/data to the block. The SPI Master does not need to know the base address of the SPI Slave in EC.

The Serial Peripheral Interface (SPI) Slave can be accessed from the internal embedded controller (EC) and SPI Master. Once the Mask EC Registers bit is set in the System Configuration Register then only SPI Slave Status Register, SPI Master-to-EC Mailbox Register, EC-to-SPI Master Mailbox Register and SPI Interrupt Enable Register are accessible to the SPI Master with read/write access. All other SPI Slave registers listed in table Table 11-8, "Register Summary" are Read-Only. However all registers are Readable/Writable by EC.

Note: SPI Master only has read access to these register when lock bit is set or is totally hidden if Mask EC Registers is set.

Note: Serial Peripheral Interface (SPI) Slave should be enabled by firmware only after the PLL has locked. Refer Section 4.7.4, "Waking the Chip From Sleeping State" for details.

Note: The SPI Master has to wait for the wake up timing requirements from heavy sleep after asserting SLV_SPI_CS# and before initiating the read / write transfer. This time is required for the PLL to come up and lock and all blocks to be functioning properly. Refer Section 4.7.4, "Waking the Chip From Sleeping State" for chip sleep states.

Note: Setting Mask EC Registers bit in the System Configuration Register is must before enabling the SPI Slave module from security perspective.

TABLE 11-8: REGISTER SUMMARY

Offset	Register Name
00h	SPI Communication Configuration Register
04h	SPI Slave Status Register
08h	SPI EC Status Register
0Ch	SPI Interrupt Enable Register
10h	EC Interrupt Enable Register
14h	Memory Configuration Register
18h	Memory Base Address0 Register
1Ch	Memory Write Limit0 Register
20h	Memory Read Limit0 Register
24h	Memory Base Address1 Register
28h	Memory Write Limit1 Register
2Ch	Memory Read Limit1 Register
30h	RX FIFO Host Bar Register
34h	RX FIFO Byte Counter Register
38h	TX FIFO Host Bar Register
3Ch	TX FIFO Byte Counter Register
40h	System Configuration Register
44h	SPI Master-to-EC Mailbox Register
48h	EC-to-SPI Master Mailbox Register

© 2021 Microchip Technology Inc.

Note 1: SPI Access is limited by the corresponding <Lock> bit and the <Mask> bit as follows:

- (NL-NM): “Not Locked and not Masked” has same access as EC.
- (L-NM): “Locked and not Masked” has RO access.
- (NL-M): “Not Locked and Masked” is reserved.
- (L-M): “Locked and Masked” is reserved.

2: SPI Access is limited by the only corresponding <Lock> bit as follows:

- (NL-NM): “Not Locked and not Masked” has same as EC access
- (L-NM): “Locked and not Masked” has RO access.
- (NL-M): “Not Locked and Masked” has same as EC access.
- (L-M): “Locked and Masked” has RO access.

3: SPI Access is limited by only the <Mask> bit as follows:

- (NL-NM): “Not Locked and not Masked” has RO* access.
- (L-NM): “Locked and not Masked” not applicable.
- (NL-M): “Not Locked and Masked” is reserved.
- (L-M): “Locked and Masked” is not applicable.

4: SPI Master has full access.

5: SPI Master does not have access.

6: Turn Around Time comes into picture when SPI Data have transmit followed by receive or receive followed by transmit in one command. Meaning, direction of data on the SPI bus is changing. This is the time when the current SPI data bus driver is turning of the SPI data enable.

7: Wait time is used to allow the operation to complete on a different clock domain that SPI Clock.

8: Dual Mode is not supported by SPI Slave hardware.

9: To prevent SPI Master transactions from hanging the internal bus, if the data is in less that projected, the AHB request is withdrawn and this error flag is set. if Interrupt to the SPI master is enabled, it will know the problem.

10: Programming Memory Base Address0 Register and Memory Base Address1 Register register, allows the SPI Master to access any peripheral/memory mapped area in the chip. Using this register SPI Master may access other peripherals in the EC space (Say UART or I2C or other peripherals). If those peripherals are busy for some reason, this bit will be set.

11: By using command CMD_IN_BAND_RST, SPI Master may request the SPI Slave block to be reset. All the SPI Slave registers will be reset and will need to be reprogrammed by the EC. This comes in handy when the SPI Master is out of SYNC with the SPI Slave and there is no way to recover. Also in case there is an error in the SPI Master that needs reconfiguring the SPI Slave, this command could be used.

11.9.1 SPI COMMUNICATION CONFIGURATION REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:16	Wait time This bit sets the amount of wait time in cycles before transmitting data back to master. During this wait time lower 8 bits of SPI Slave Status Register will be transmitted. Also see Note 7.	R/W Note 2	4h	RESET_SYS
15:10	Reserved	RES	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9:8	TAR Time Turn Around Time select for Quad Wire. Also see Note 6 • 0h = 1 cycle • 1h = 2 cycles • 2h = 4 cycles • 3h = 8 cycles • Other values are reserved.	R/W Note 2	0h	RESET_SYS
7:1	Reserved	RES	-	-
0	Single / Quad Wire Select • 0 = Single Wire • 1 = Quad Wire Also see Note 8.	R/W Note 2	0h	RESET_SYS

The lock register can be enabled to lock all or certain fields from SPI Master.

Note: If there are writes done to EC registers that require clock domain transfer and the wait cycles programed in the register are not sufficient, once the transaction is captured by the SPI Slave, it will go through.

Note: The SPI Slave has to respond with the 2 byte status packet for the write command after fixed wait cycles programmed in the SPI Communication Configuration Register bits [23:16]. If the transaction did not complete during this Wait time, the SPI Slave will return busy status in the status packet. Also see Non-Posted Memory (Block) and SREG Write Command Format.

Note: Please note that different commands may have different wait requirement.

11.9.2 SPI SLAVE STATUS REGISTER

The below register is for SPI Master and is implemented in the SPI Clock domain.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow If SPI Master writes more than the space in the FIFO, the FIFO will flag an overflow error and data will not be stored.	R/WC Note 2	0h	RESET_SYS
27	RX FIFO Underflow If the SPI Slave reads RX FIFO when it is empty, RX FIFO Underflow flag will be set. This condition will never happen under normal situation.	R/WC Note 2	0h	RESET_SYS
26	TX FIFO Overflow If Master doesn't read all of the data it requested from the posted read block cycle, than data will still be left in the FIFO. This will cause misalignment with the following transactions and a new read cycle can cause overflow.	R/WC Note 2	0h	RESET_SYS
25	TX FIFO Underflow If Master reads more than what is in FIFO, FIFO will flag an underflow error and the data returned will just be the last valid pointer value.	R/WC Note 2	0h	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
24	RX FIFO Size Error If the Master terminates a command early then stated in the command, an error flag shut down request signal to ARM Bus. If the Master provides more data than stated in the command, then SPI Slave ignored and continue transaction. this may mean that SPI Slave is taking in garbage value. Also see Note 9.	R/WC Note 2	0h	RESET_SYS
23	DV_Busy If the Master requested a transaction whose destination is busy the request is ignored. Should use the poll or wait for interrupts. Also see Note 10.	R/WC Note 2	0h	RESET_SYS
22	Undefined Command This flag is set when unknown Command is received from the SPI Master.Command is ignored and Status bit set. If the interrupt is enabled, it will also trigger interrupt.	R/WC Note 2	0h	RESET_SYS
21	ARM BUS Error This bit is set when there is an error on the internal bus. If there is an error in the internal AHB, this error is set. This will indicate to SPI Master to reset and retry the transaction again. If the error persists, SPI Master may have accesses unavailable space.	R/WC Note 2	0h	RESET_SYS
20	Out of Limit 1 Error Address requested out of range or request when the BAR is disabled. Please see Memory Base Address0 Register, Memory Write Limit0 Register and Memory Read Limit0 Register. These registers set the address range	R/WC Note 2	0h	RESET_SYS
19	Out of Limit 0 Error Address requested out of range or request when the BAR is disabled. Please see Memory Base Address1 Register, Memory Write Limit1 Register and Memory Read Limit1 Register. These registers set the address range.	R/WC Note 2	0h	RESET_SYS
18	TX FIFO Reset Done This bit is set after the SPI Master initiates a reset and the reset has been completed on the TX FIFO. This bit indicates that FIFO is cleared. If The SPI Master issues reset, it should wait for the status of the TX FIFO Reset Done bit to be set before continuing with any other command.	R/WC Note 2	0h	RESET_SYS
17	RX FIFO Reset Done This bit is set after the SPI Master initiates a reset and the reset has been completed on the RX FIFO. This bit indicates that FIFO is cleared. If The SPI Master issues reset command, it should wait for the status of the RX FIFO Reset Done bit to be set before continuing with any other command.	R/WC Note 2	0h	RESET_SYS
16	SPI Master Requested Reset This bit is set when the SPI Master Requested a Configuration Reset. Also see Note 11.	R/WC Note 2	0h	RESET_SYS
15	OBF Flag This bit is set when the EC writes to the internal SPI Slave Buffer signaling there is data for the SPI Master to read.	R	0h	RESET_SYS
14	IBF Flag This bit is set when the Host writes to the Input Buffer signaling there is data for the EC to read.	R	0h	RESET_SYS

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
13:12	Reserved	RES	-	-
11	TX FIFO Full This bit is set when the TX FIFO is full and SPI Master needs to read the data.	R	0h	RESET_SYS
10	TX FIFO Empty This bit is set when the internal FIFO in SPI Slave is empty. If this bit is not set, it means there is still data left in the TX FIFO. SPI Master needs to read the remaining data.	R	1h	RESET_SYS
9	RX FIFO Full This bit is set when RX FIFO is full of data to be written to Memory. SPI Master should not initiate any more data write to this FIFO until there is space available to shore the full data packet.	R	0h	RESET_SYS
8	RX FIFO Empty This bit is set when the RX FIFO is empty. The SPI Master may initiate new write transfers to the SPI Slave.	R	1h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req If this bit is set, then something in the high 16-bit of status register is set and needs to be checked. SPI Master should take action to clear this bit.	R	0h	RESET_SYS
5	SREG Trans Busy This bit is set when an SREG transaction is currently being processed. Also see Note 3	R	0h	RESET_SYS
4	Memory Read Busy This bit is set when an Memory Read transaction is currently being processed. Also see Note 4.	R	0h	RESET_SYS
3	Memory Write Busy This bit is set when an Memory Write transaction is currently being processed. Also see Note 5.	R	0h	RESET_SYS
2	Reserved	RES	-	-
1	Memory Read Done When the ARM BUS side has fully finished writing the last written DWord to the FIFO for a set of data read from Memory. This register bit is cleared by writing to this register. Also see Note 6.	R/WC Note 2	0h	RESET_SYS
0	Memory Write Done This bit is set when the ARM BUS side has fully finished the last transaction from the FIFO to write the data to Memory. This register bit is cleared by writing to this register.	R/WC Note 2	0h	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
Note 1: This register is accessible by the SPI Master only in Advanced Mode.				
2: Upon reset of the SPI Slave block, RX FIFO Reset Done and TX FIFO Reset Done bits will get asserted only after several SPI Clocks have been received as the FIFOs require SPI Clock to have a complete reset.				
3: In SPI slave, only one transaction Read or Write can happen at a time. Until the previous transfer is not over, SPI Master should not initiate a new transfer.				
4: Memory Read Busy and Memory Read Done are completely opposite to each other. When one bit is set the other will be cleared.				
5: Memory Write Busy and Memory Write Done are completely opposite to each other. When one bit is set the other will be cleared.				
6: Please refer Section 11.10.1.1, "Non-Posted Memory (Block) and SREG Write Command Format", Section 11.10.1.2, "Non-Posted Memory (Block) and SREG Read Command Format", Section 11.10.1.3, "Posted Memory Write Command Format" and Section 11.10.1.4, "Posted Memory Read Command Format" to know the list of commands that allow SPI Master to initiate Posted Vs Non-Posted memory Read/Write request to SPI Slave.				

11.9.3 SPI EC STATUS REGISTER

The below register is for EC firmware and is implemented in the 48MHz (EC Clock) domain.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow If SPI Master writes more than the space in the FIFO, the FIFO will flag an overflow error and data will not be stored.	R/WC	0h	RESET_SYS
27	RX FIFO Underflow If the SPI Slave reads RX FIFO when it is empty, RX FIFO Underflow flag will be set. This condition will never happen under normal situation.	R/WC	0h	RESET_SYS
26	TX FIFO Overflow If Master doesn't read all of the data it requested from the posted read block cycle, than data will still be left in the FIFO. This will cause misalignment with the following transactions and a new read cycle can cause overflow.	R/WC	0h	RESET_SYS
25	TX FIFO Underflow If Master reads more than what is in FIFO, FIFO will flag an underflow error and the data returned will just be the last valid pointer value.	R/WC	0h	RESET_SYS
24	RX FIFO Size Error If the Master terminates a command early then stated in the command, an error flag shut down request signal to ARM Bus. If the Master provides more data than stated in the command, then SPI Slave ignored and continue transaction. this may mean that SPI Slave is taking in garbage value. Also see Note 9.	R/WC	0h	RESET_SYS

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
23	DV_Busy If the Master requested a transaction whose destination is busy the request is ignored. Should use the poll or wait for interrupts. Also see Note 10.	R/WC	0h	RESET_SYS
22	Undefined Command This bit is set when unknown Command is received from the SPI Master.Command is ignored and Status bit set. If the interrupt is enabled, it will also trigger interrupt.	R/WC	0h	RESET_SYS
21	ARM BUS Error This bit is set when there is an error on the internal bus. If there is an error in the internal AHB, this error is set. This will indicate to SPI Master to reset and retry the transaction again. If the error persists, SPI Master may have accesses unavailable space.	R/WC	0h	RESET_SYS
20	Out of Limit 1 Error Address requested out of range or request when the BAR is disabled. Please see Memory Base Address0 Register, Memory Write Limit0 Register and Memory Read Limit0 Register. These registers set the address range	R/WC	0h	RESET_SYS
19	Out of Limit 0 Error Address requested out of range or request when the BAR is disabled. Please see Memory Base Address1 Register, Memory Write Limit1 Register and Memory Read Limit1 Register. These registers set the address range.	R/WC	0h	RESET_SYS
18	TX FIFO Reset Done This bit is set after the SPI Master initiates a reset and the reset has been completed on the TX FIFO. This bit indicates that FIFO is cleared. If The SPI Master issues reset, it should wait for the status of the TX FIFO Reset Done bit to be set before continuing with any other command.	R/WC	0h	RESET_SYS
17	RX FIFO Reset Done This bit is set after the SPI Master initiates a reset and the reset has been completed on the RX FIFO. This bit indicates that FIFO is cleared. If The SPI Master issues reset command, it should wait for the status of the RX FIFO Reset Done bit to be set before continuing with any other command.	R/WC	0h	RESET_SYS
16	SPI Master Requested Reset This bit is set when the SPI Master Requested a Configuration Reset. Also see Note 11.	R/WC	0h	RESET_SYS
15	OBF Flag This bit is set when the EC writes to the internal SPI Slave Buffer signaling there is data for the SPI Master to read.	R	0h	RESET_SYS
14	IBF Flag This bit is set when the Host writes to the Input Buffer signaling there is data for the EC to read.	R	0h	RESET_SYS
13:12	Reserved	RES	-	-
11	TX FIFO Full This bit is set when the TX FIFO is full and SPI Master needs to read the data.	R	0h	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TX FIFO Empty This bit is set when the internal FIFO in SPI Slave is empty. If this bit is not set, it means there is still data left in the TX FIFO. SPI Master needs to read the remaining data.	R	1h	RESET_SYS
9	RX FIFO Full This bit is set when RX FIFO is full of data to be written to Memory. SPI Master should not initiate any more data write to this FIFO until there is space available to shore the full data packet.	R	0h	RESET_SYS
8	RX FIFO Empty This bit is set when the RX FIFO is empty. The SPI Master may initiate new write transfers to the SPI Slave.	R	1h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req If this bit is set, then something in the high 16-bit of status register is set and needs to be checked. SPI Master should take action to clear this bit.	R	0h	RESET_SYS
5	SREG Trans Busy This bit is set when an SREG transaction is currently being processed. Also see Note 3	R	0h	RESET_SYS
4	Memory Read Busy This bit is set when an Memory Read transaction is currently being processed. Also see Note 4.	R	0h	RESET_SYS
3	Memory Write Busy This bit is set when an Memory Write transaction is currently being processed. Also see Note 5.	R	0h	RESET_SYS
2	Reserved	RES	0h	RESET_SYS
1	Memory Read Done When the ARM BUS side has fully finished writing the last written DWord to the FIFO for a set of data read from Memory. This register bit is cleared by writing to this register. Also see Note 6.	R/WC	0h	RESET_SYS
0	Memory Write Done This bit is set when the ARM BUS side has fully finished the last transaction from the FIFO to write the data to Memory. This register bit is cleared by writing to this register.	R/WC	0h	RESET_SYS

11.9.4 SPI INTERRUPT ENABLE REGISTER

The below register is for SPI Master and is implemented in the SPI Clock domain. This register controls the assertion of SLV_SPI_MSTR_INT to SPI Master.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
27	RX FIFO Underflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
26	TX FIFO Overflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
25	TX FIFO Underflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
24	RX FIFO Size Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
23	DV_Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
22	Undefined Command: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
21	ARM BUS Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
20	Out of Limit 1 Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
19	Out of Limit 0 Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
18	TX FIFO Reset Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
17	RX FIFO Reset Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
16	SPI Master Requested Reset: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
15	OBF Flag: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
14	Reserved	RES	-	-
13	TM SPI Clock Count Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
12	Reserved	RES Note 2	-	-
11	TX FIFO Full: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
10	TX FIFO Empty: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
9	RX FIFO Full: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
8	RX FIFO Empty: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
5	SREG Trans Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
4	Memory Read Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
3	Memory Write Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	Reserved	RES Note 2	0h	RESET_SYS
1	Memory Read Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
0	Memory Write Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
Note:	This register may be accessible by the SPI Master only in Advanced Mode.			

11.9.5 EC INTERRUPT ENABLE REGISTER

The below register is for EC firmware and is implemented in the 48MHz (EC Clock) domain. This register controls the assertion of SPI_EC_INTERRUPT to EC.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
27	RX FIFO Underflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
26	TX FIFO Overflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
25	TX FIFO Underflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
24	RX FIFO Size Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
23	DV_Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
22	Undefined Command: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
21	ARM BUS Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
20	Out of Limit 1 Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
19	Out of Limit 0 Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
18	TX FIFO Reset Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
17	RX FIFO Reset Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
16	SPI Master Requested Reset: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
15	OBF Flag: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
14	IBF Flag: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
13	TM SPI Clock Count Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set	R/WC Note 3	0h	RESET_SYS
12	Reserved	RES	-	-
11	TX FIFO Full: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
10	TX FIFO Empty: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
9	RX FIFO Full: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
8	RX FIFO Empty: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
5	SREG Trans Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
4	Memory Read Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
3	Memory Write Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
2	Reserved	RES	0h	RESET_SYS
1	Memory Read Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
0	Memory Write Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS

11.9.6 MEMORY CONFIGURATION REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	BAR 1 Enable: Enables Region 1	R/W Note 1	0h	RESET_SYS
0	BAR 0 Enable: Enables Region 0	R/W Note 1	0h	RESET_SYS

11.9.7 MEMORY BASE ADDRESS0 REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Base Address for Region 0 **Note:** Base Address for Region 0 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

© 2021 Microchip Technology Inc.

11.9.8 MEMORY WRITE LIMIT0 REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Write Limit for Region 0 **Note:** Write Limit for Region 0 is DWORD aligned.	R/W Note 1	0	RESET_SYS
1:0	Reserved	RES	-	-

11.9.9 MEMORY READ LIMIT0 REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Read Limit for Region 0 **Note:** Read Limit for Region 0 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

Note: Memory Base Address0 Register register controls the region of memory address space within EC accessible from SPI Master. Memory Write Limit0 Register controls the region of address space within the Memory Base Address0 Register that is writable and Memory Read Limit0 Register controls the region of address space within the Memory Base Address0 Register that is readable by SPI Master.

Note: Application should never set the Memory Write Limit0 Register or Memory Read Limit0 Register value greater than the limit available.

Note: If it is desired that a region of memory be dedicated only for SPI Master read, Memory Write Limit0 Register must be set to all zeros. If it is desired that a region of memory be dedicated only for SPI Master Write, Memory Read Limit0 Register must be set to all zeros.

11.9.10 MEMORY BASE ADDRESS1 REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Base Address for Region 1 **Note:** Base Address for Region 1 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

11.9.11 MEMORY WRITE LIMIT1 REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Write Limit for Region 1 **Note:** Write Limit for Region 1 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

11.9.12 MEMORY READ LIMIT1 REGISTER

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Read Limit for Region 1 **Note:** Read Limit for Region 1 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

Note: SPI Slave expects that all transfer from the SPI Master will be terminated at the buffer boundary. There is no address wrap around implemented in the SPI Slave. Once the address hits the final value allowed through Memory Write Limit0 Register / Memory Read Limit0 Register or Memory Write Limit1 Register / Memory Read Limit1 Register, inhibit_wrap_around internal signal will assert and the RX FIFO Byte Counter Register and TX FIFO Byte Counter Register will stop counting.

11.9.13 RX FIFO HOST BAR REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	RX FIFO Bar Latest offset address requested by the SPI Master for a write transfer. This register gets set for a new transaction request.	R	0h	RESET_SYS

11.9.14 RX FIFO BYTE COUNTER REGISTER

Offset	34h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
14:0	RX FIFO Byte Count Number of Bytes written through the AHB transfer. This register gets cleared for every new request	R	0h	RESET_SYS

11.9.15 TX FIFO HOST BAR REGISTER

Offset	38h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	TX FIFO Bar: Latest offset address requested by the SPI Master for a read transfer. This register gets set for a new transaction request.	R	0h	RESET_SYS

Note: The RX FIFO Host Bar Register and TX FIFO Host Bar Register may be helpful in debugging.

© 2021 Microchip Technology Inc.

11.9.16 TX FIFO BYTE COUNTER REGISTER

Offset	3Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
14:0	TX FIFO Byte Count: Number of Bytes written through the AHB transfer. This register gets cleared for every new request	R	0h	RESET_SYS

11.9.17 SYSTEM CONFIGURATION REGISTER

Note: EC access to registers remains the same in all configurations. However the SPI Master access can be changed depending on the configurations set in this register.

Note: The System Configuration Register is a read only register from SPI Master. It can be read and written by the EC at all times. The Lock bits in this register only define the type of access for the SPI Master.

Note: Any read of the SPI Slave registers when Mask EC Registers bit is set, will always return 0h value to the SPI Master.

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:20	Reserved	RES	-	-
19	EC Data Available Notification to TX FIFO Engine that data is available for AHB Transfer. This register but is cleared by Hardware at the end of the transaction, with SLV_SPI_CS# de-assertion. This register bit is mainly for debug.	R/WC Note 3	0h	RESET_SYS
18	Simple Mode: Enable SPI Slave Simple Mode operation 0 = Advanced Mode 1 = Simple Mode	R/W Note 3	0h	RESET_SYS
17	Mask EC Registers Mask EC registers Mask EC Registers from SPI Master. All the register are neither readable now writable from SPI Master.	R/W Note 3	0h	RESET_SYS
16	Activate SPI Slave Block Enabled / Disabled 0 = Disable 1 = Enable	R/W Note 3	0h	RESET_SYS
15:10	Reserved	RES	-	-
9	Reserved	RES	-	-
8	Reserved	RES	-	-
7	Lock Mem Bar1: Lock writes to Region 1 Addresses from SPI Master. 0 = Unlocked 1 = Locked	R/W Note 3	1h	RESET_SYS
6	Lock Mem Bar0: Lock writes to Region 0 Addresses from SPI Master 0 = Unlocked 1 = Locked	R/W Note 3	1h	RESET_SYS

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	Lock SPI Int En: Lock SPI interrupt enable register from being modified by SPI Master 0 = Unlocked 1 = Locked	R/W Note 3	0h	RESET_SYS
4	Lock SPI Stats: Lock write access to SPI Status field from SPI Master. 0 = Unlocked 1 = Locked	R/W Note 3	0h	RESET_SYS
3	Lock Wait Cycles: Lock Wait time register bits from being modified by SPI Master. 0 = Unlocked 1 = Locked	R/W Note 3	0h	RESET_SYS
2	Lock Tar Time: Lock TAR Time register bits from being modified by SPI Master. 0 = Unlocked 1 = Locked	R/W Note 3	0h	RESET_SYS
1	Lock Quad / Single Write Mode: Lock Single / Quad Wire Select register bits from being modified by SPI Master. 0 = Unlocked 1 = Locked	R/W Note 3	0h	RESET_SYS
0	Soft reset Soft reset for entire SPI Slave Block. This bit is self clearing. 0 = Normal operation 1 = Soft Reset the block	WO Note 5	0h	RESET_SYS

11.9.18 SPI MASTER-TO-EC MAILBOX REGISTER

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	SPI Master to EC Write only register for the Host. When data is written to this register the IBF Flag is set. EC can read the data and writes of 0xFFFF_FFFF will clear this register. Any form of read will clear the flag for this register.	R/WC Note 4	0h	RESET_SYS

11.9.19 EC-TO-SPI MASTER MAILBOX REGISTER

Offset	48h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	EC to SPI Master This is a Read only register for the SPI Master. When data is written to this register the OBF Flag is set. SPI Master can read the data and writes of 0xFFFF_FFFF will clear this register, also clearing the flag. Any form of read will clear the flag for this register.	R/WC Note 4	0h	RESET_SYS

Note: Because any form of read (8/16/32 bit read) will clear the flag – it is necessary for the EC and the external FW of which the SPI Master resides agrees upon a protocol.

© 2021 Microchip Technology Inc.

11.10 Commands Supported

The list of commands supported by SPI slave is given below in the following table.

TABLE 11-9: SPI COMMANDS

Command Name	Code	Description
CMD_IN_BAND_RST	FFh	In Band Reset.
CMD_UNDEF_DWORD_W	01h	Undefined Size DWord Write
CMD_UNDEF_BYTE_W	02h	Undefined Size Byte Write
CMD_UNDEF_DWORD_R	05h	Undefined Size DWord Read
CMD_UNDEF_BYTE_R	06h	Undefined Size Byte Read
CMD_RST_RX_FIFO	12h	Reset RX FIFO pointers
CMD_RST_TX_FIFO	14h	Reset TX FIFO pointers
CMD_RST_RXTX_FIFO	16h	Reset RX and TX FIFO pointers
CMD_EXT_REG_W8	41h	External Register Bank 8 bit write
CMD_EXT_REG_R8	45h	External Register Bank 8 bit read
CMD_SREG_W8	09h	SPI Slave Register 8 bit Write
CMD_SREG_W16	0Ah	SPI Slave Register 16 bit Write
CMD_SREG_W32	0Bh	SPI Slave Register 32 bit Write
CMD_SREG_R8	0Dh	SPI Slave Register 8 bit Read
CMD_SREG_R16	0Eh	SPI Slave Register 16 bit Read
CMD_SREG_R32	0Fh	SPI Slave Register 32 bit Read
CMD_MEM_W8	21h	Standalone 8 bit Memory Write
CMD_MEM_W16	22h	Standalone 16 bit Memory Write
CMD_MEM_W32	23h	Standalone 32 bit Memory Write
CMD_MEM_R8	25h	Standalone 8 bit Memory Read
CMD_MEM_R16	26h	Standalone 16 bit Memory Read
CMD_MEM_R32	27h	Standalone 32 bit Memory Read
CMD_RD_SNGL_FIFO8	28h	Standalone 8 bit Memory Read FIFO
CMD_RD_SNGL_FIFO16	29h	Standalone 16 bit Memory Read FIFO
CMD_RD_SNGL_FIFO32	2Bh	Standalone 32 bit Memory Read FIFO
CMD_RD_SNGL_FIFO8_FSR	68h	8 bit Memory Read FIFO with Status
CMD_RD_SNGL_FIFO16_FSR	69h	16 bit Memory Read FIFO with Status
CMD_RD_SNGL_FIFO32_FSR	6Bh	32 bit Memory Read FIFO with Status
CMD_POLL_LOW	2Ch	Read lower 16 bits of the Status Register
CMD_POLL_HIGH	2Dh	Read higher 16 bits of the Status Register
CMD_POLL_ALL	2Fh	Read all 32 bits of the Status Register
CMD_EXTEND	6Ch	Declare Second Command Byte
CMD_MEM_BLK_W	80h-87h	Block 1-8 D word Memory Write
CMD_MEM_BLK_R	A0h-A7h	Block 1-8 D word Memory Read
CMD_RD_BLK_FIFO	C0h-C7h	Block 1-8 D word Read FIFO
CMD_BLK_RD_FIFO_FSR	E0h-E7h	Block 1-8 D word Read FIFO with status

Note: SPI Master should use CMD_IN_BAND_RST command when something went really wrong with the communication between the Master and Slave, i.e. error occurred when setting the configuration and both SPI Master and Slave are out of sync. SPI Master needs to poll for Soft reset bit to be cleared before continuing with any further command.

Note: All the commands that use DWORD (32Bit) transfers will utilize the internal AHB bus bandwidth better. It could be Quad/Single Wire based on the Single / Quad Wire Select setting.

Note: For CMD_MEM_BLK_W, CMD_MEM_BLK_R, CMD_RD_BLK_FIFO, CMD_BLK_RD_FIFO_FSR commands last nibble 0-7 represent the 1 to 8 D Word operation. where 0 represents 1 D word and 7 represents 8 D word operation. This is equivalent to saying that there are 8 commands for each of these commands.

Note: For CMD_UNDEF_DWORD_W, CMD_UNDEF_BYTE_W, CMD_UNDEF_DWORD_R and CMD_UNDEF_BYTE_R commands, the SPI Master first needs to issue CMD_EXTEND command. This will inform the SPI Slave that the following command will have two parameters preset in the command. The SPI slave expects only one parameter in the usual case along with the command.

Note: Commands CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32 and CMD_MEM_BLK_W are be used to generate AHB write transaction in the EC, pointed by Memory Base Address0 Register and Memory Base Address1 Register and the offset address received from the master.

Note: Commands CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32 and CMD_MEM_BLK_R are be used to generate AHB read transaction in the EC, pointed by Memory Base Address0 Register and Memory Base Address1 Register and the offset address received from the master.

Note: For CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32, CMD_MEM_BLK_W, CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32 and CMD_MEM_BLK_R the address bit 15 received from the SPI Master determines to which memory base address region the transaction is targeted. If bit 15 of the address is 0b, the transaction if for Memory Base Address0 Register and if bit 15 of the address is 1b, the transaction is for Memory Base Address1 Register.

Note: Following SPI Slave module registers could be accessed by the SPI Master using CMD_SREG_* commands when the registers are not locked. SPI Communication Configuration Register, Memory Base Address0 Register, Memory Write Limit0 Register, Memory Read Limit0 Register, Memory Base Address1 Register, Memory Write Limit1 Register, Memory Read Limit1 Register, RX FIFO Host Bar Register, RX FIFO Byte Counter Register, TX FIFO Host Bar Register, TX FIFO Byte Counter Register, System Configuration Register. Here * could have one of the following values W8, W16, W32, R8, R16 and R32.

Note: All the commands that return status, like CMD_MEM_W8, etc will return the value in lower 2 bytes of the SPI Slave Status Register.

11.10.1 COMMAND FORMAT

This section lists the command format for each command category

11.10.1.1 Non-Posted Memory (Block) and SREG Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-2 for non-posted writes. CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32, CMD_MEM_BLK_W, CMD_SREG_W8, CMD_SREG_W16 and CMD_SREG_W32 commands can be used for this type of transfer.

FIGURE 11-2: NON-POSTED MEMORY (BLOCK) AND SREG WRITE COMMAND FORMAT

For these commands, the SPI Slave sends the write command status at the end of the transaction. These commands are ideal for cases where the SPI Master wants to see the status without issuing another command to confirm the transfer status.

© 2021 Microchip Technology Inc.

Note:	Wait States are used with commands that involve clock domain transfer of data from SLV_SPI_SCLK domain to 48MHz clock and Visa Versa. During wait state, SPI Slave will return the lower 8-bit of status register which contain the status of the transaction.
Note:	Wait time should be programmed to a value that will allow transactions to get done/error response to come, but with minimal latency. This is dependent on the relation between the SLV_SPI_SCLK and 48MHz internal clocks.
Note:	The status bits during wait will be lower 8 bits of SPI Slave Status Register and the final status will be lower 2 bytes (16 bits) of SPI Slave Status Register.

11.10.1.2 Non-Posted Memory (Block) and SREG Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-3 for non-posted reads. CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32, CMD_MEM_BLK_R, CMD_SREG_R8, CMD_SREG_R16 and CMD_SREG_R32 commands can be used for this type of transfer.

FIGURE 11-3: NON-POSTED MEMORY (BLOCK) AND SREG READ COMMAND FORMAT

For these commands, the SPI Slave sends the status and the read data. These commands are ideal for cases where the SPI Master wants to see the status without issuing another command to confirm the transfer status.

11.10.1.3 Posted Memory Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-4 for posted memory writes. CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32 and CMD_MEM_BLK_W commands can be used for this type of transfer. The SPI Master needs to end the command by de-asserting the SLV_SPI_CS# after the command, address and data is transfered. The early termination of the command indicates that the current transfer is Posted transfer to the SPI slave. The SPI Master can do other work and later check the status of the transfer via the CMD_POLL_LOW, CMD_POLL_HIGH and CMD_POLL_ALL command.

FIGURE 11-4: POSTED MEMORY WRITE COMMAND FORMAT

11.10.1.4 Posted Memory Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-5 for posted memory read.CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32 and CMD_MEM_BLK_R commands can be used for this type of transfer. The SPI Master needs to end the command by de-asserting the SLV_SPI_CS# after the command and address is transfered. The early termination of the command indicates that the current transfer is Posted transfer to the SPI slave. The SPI Master can do other work and later check the status of the transfer via the CMD_POLL_LOW, CMD_POLL_HIGH and CMD_POLL_ALL command. Once the status indicates that data is available (via TX FIFO Full or TX FIFO Empty), SPI Master initiates a FIFO read command to get the data. See Section 11.10.1.7, FIFO Read Command Format for details.

FIGURE 11-5: POSTED MEMORY READ COMMAND FORMAT

11.10.1.5 Undefined Size Memory Write Command Format

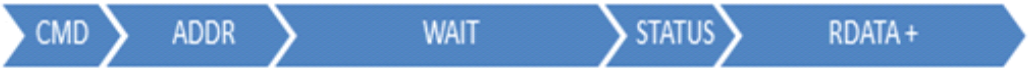
The SPI Master should expect the command and data in below order as shown in Figure 11-6 for Undefined size memory write transfer. CMD_EXTEND, CMD_UNDEF_DWORD_W and CMD_UNDEF_BYTE_W commands can be used for this type of transfer. The SPI Master needs to end the Undefined Size Memory Write transfer by de-asserting the SLV_SPI_CS#. There will be no status for this transfer provided to the SPI Master, however the SPI Master may use CMD_POLL_LOW, CMD_POLL_HIGH or CMD_POLL_ALL command to know the status of the transfer.

FIGURE 11-6: UNDEFINED SIZE MEMORY WRITE COMMAND FORMAT

11.10.1.6 Undefined Size Memory Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-7 for Undefined size memory read transfer. CMD_EXTEND, CMD_UNDEF_DWORD_R and CMD_UNDEF_BYTE_R commands can be used for this type of transfer. The SPI Master needs to end the Undefined Size Memory Read transfer by de-asserting the SLV_SPI_CS#. The status returned by the Serial Peripheral Interface (SPI) Slave informs whether there is data in the FIFO available for the SPI Master to read.

FIGURE 11-7: UNDEFINED SIZE MEMORY READ COMMAND FORMAT



11.10.1.7 FIFO Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-8 for FIFO Read transfer. CMD_RD_SNGL_FIFO8, CMD_RD_SNGL_FIFO16, CMD_RD_SNGL_FIFO32 and CMD_RD_BLK_FIFO commands can be used for this type of transfer. This command is used along with Posted Memory read command. See section Section 11.10.1.4, Posted Memory Read Command Format for more information.

FIGURE 11-8: FIFO READ COMMAND FORMAT

11.10.1.8 FIFO Read with Status Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-9 for FIFO Read with Status transfer. CMD_RD_SNGL_FIFO8_FSR, CMD_RD_SNGL_FIFO16_FSR, CMD_RD_SNGL_FIFO32_FSR and CMD_BLK_RD_FIFO_FSR commands can be used for this type of transfer. This command is used along with the posted read command and helps avoid constant polling the SPI Slave via CMD_POLL_LOW, CMD_POLL_HIGH or CMD_POLL_ALL commands. The SPI Master may issue the FIFO Read with Status Command read the data if data is available (as indicated in the Status) or terminate the transaction after status read, if data is not available.

FIGURE 11-9: FIFO READ WITH STATUS COMMAND FORMAT

© 2021 Microchip Technology Inc.

11.10.1.9 Poll Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-10 for Poll transfer. CMD_POLL_LOW, CMD_POLL_HIGH or CMD_POLL_ALL commands can be used for this type of transfer. This command will immediately return the contents of SPI Slave Status Register. This command should be used by the SPI Master before issuing the command of the same type, check for errors that may have occurred with any previous transaction or check for data availability in case of posted transactions.

FIGURE 11-10: POLL COMMAND FORMAT

11.10.1.10 External Register Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-11 for External Register write transfer. CMD_EXT_REG_W8 command can be used for this type of transfer. This command is used to write to the SPI Slave register bank. The status of this command will be immediately returned back to the SPI Master. This type of access is used to write to registers in the SPI clock domain.

FIGURE 11-11: EXTERNAL REGISTER WRITE COMMAND FORMAT

11.10.1.11 External Register Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 11-12 for External Register read transfer. CMD_EXT_REG_R8 command can be used for this type of transfer. This command is used to read the register bank. The status of this command will be immediately returned back to the SPI Master. This type of access is used to read registers in the SPI clock domain.

FIGURE 11-12: EXTERNAL REGISTER READ COMMAND FORMAT

11.10.1.12 Simple Mode

The SPI Master should expect the command and data in below order as shown in Figure 11-13 for External Register read transfer.

FIGURE 11-13: SIMPLE MODE

11.10.1.13 Reset Commands Format

The SPI Master should expect the Reset command behavior as shown in Figure 11-14. CMD_RST_RX_FIFO, CMD_RST_TX_FIFO and CMD_RST_RXTX_FIFO commands show the behavior shown in Figure 11-14. There should be at least 1 dummy cycle of SLV_SPI_SCLK clock for the reset operation to complete.

FIGURE 11-14: RESET COMMAND FORMAT

11.11 Examples

This section shows an example usage of this block. The example algorithm for initialization from EC (EC Initialization) and SPI Master (SREG Accesses by SPI Master) are discussed in this section

11.11.1 EC INITIALIZATION

1. SPI Slave configured as a Bridge
    a. Configure EC Interrupt Enable Register and SPI Interrupt Enable Register
    b. Enable SPI Slave Block
        i. configure System Configuration Register with 32'h0001_0000
            1. Activate Slave (bit 16) and unlocks write access to registers
2. SPI Slave configured in non bridge mode:
    a. Configure Memory Configuration Register, Memory Base Address0 Register and Memory Base Address1 Register
    b. Configure SPI Communication Configuration Register
    c. Configure EC Interrupt Enable Register and SPI Interrupt Enable Register
    d. Enable SPI Slave Block
        i. Write System Configuration Register with 32'h0001_04CE
            1. Activate Slave (bit 16) and lock all write access
    b. EC Configures GPIO's
3. SPI Slave configured in non bridge mode (solely prohibiting Memory Configuration write accesses from SPI Master):
    a. Configure Memory Configuration Register, Memory Write Limit0 Register, Memory Read Limit0 Register, Memory Base Address0 Register, Memory Write Limit1 Register, Memory Read Limit1 Register and Memory Base Address1 Register.
    b. Configure SPI Communication Configuration Register
    c. Enable SPI Slave Block
        i. Write System Configuration Register with 32'h0001_00C0
            1. Activate Slave (bit 16) and lock all write access
    b. EC Configures GPIO's

Note:	In 3, the SPI Master can still access/write SPI Communication Configuration Register and SPI Interrupt Enable Register.
Note:	It is recommended not to change the configuration of the SPI Slave register while the block is active/enabled via System Configuration Register bit 16 (Activate).

4. If SPI Master does not have any access to EC registers (this completely hides the Memory Configuration Register (other registers as well) ie. the SPI Master will not be able to read/write the base address or limits):
    a. Write Memory Configuration Register
    b. Configure EC Interrupt Enable Register and SPI Interrupt Enable Register
    c. Enable SPI Slave Block
        i. Write System Configuration Register with 32'h0003_0xxx
            1. Activate Slave (bit 16) and lock all write access

Note:	EC should make any configurations to Memory Register while block is not activated and as long as Mask EC Registers is set to confirm the SPI Master does not have access to these registers
Note:	In 4, the SPI Master still has access to write its own SPI Communication Configuration Register. See Note 2 for details about register access when Mask bit is set.

© 2021 Microchip Technology Inc.

11.11.2 SREG ACCESSES BY SPI MASTER

Note: After any reset to the Serial Peripheral Interface (SPI) Slave block, both EC and SPI Master need to clear RX FIFO RESET DONE and TX FIFO RESET DONE SPI EC Status Register and SPI Slave Status Register.

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 or CMD_SREG_W16 or CMD_SREG_W32 command with address 0x04 (access to SPI EC Status Register) and data 0x06.
2. Insert 4 Wait Cycles
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty

Note: When SPI Master is allowed to change communication Interconnect Configuration the Wait time,TAR Time may be reprogrammed by SPI Master, otherwise EC will have to set configuration through register interface.

Initialization of SPL Slave from SPI Master: Changing from 1 Turn around cycle (Current Default) to 4 Turn around cycle & 4 (Current Default) Wait Bytes to 12 Wait Bytes

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 or CMD_SREG_W16 or CMD_SREG_W32 command with address 0x00 (access to SPI Communication Configuration Register) and data 0x000C_0200 (Still in Single Wire).
2. Insert 4 Wait Cycles
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty. One may get SREG Trans Busy if current Wait Cycles is not large enough
4. Next transaction with be 4 Cycles TAR/Dummy and 12 Bytes of Wait Cycles

Initialization of SPI Slave from SPI Master: Change from Single Wire Transfer to Quad Wire

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 command with address 0x00 (access to SPI Communication Configuration Register) and data 0x01.
2. Insert 4 TAR/Dummy and 12 wait cycles and then read the Status back from SPI Slave.
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty.
4. Next transaction will use quad wire.

11.11.3 MEMORY WRITE BY SPI MASTER

1. Initiate a transaction from SPI Master with either CMD_MEM_W8 or CMD_MEM_W16 or CMD_MEM_W32 or CMD_MEM_BLK_W command with address 0x0100 and the required data bytes.
2. Insert 4 TAR/Dummy and 12 wait cycles.
3. Read the 2 byte Status back and check that all bits should be cleared except TX FIFO empty, Memory Write done and Memory Write Busy (RX FIFO not empty).

If Memory Write Busy bit is set: Poll until Done then clear bit in Status Register

1. Initiate a transaction from SPI Master with either CMD_POLL_LOW or CMD_POLL_ALL command.
2. Insert 4 TAR/Dummy
3. Read the SPI Slave Status Register back and check if Memory Write Busy bit is cleared.
4. If Memory Write Busy bit is cleared move to the next set of command, else repeat this loop.
5. Write to SPI Slave Status Register to clear the Memory Write Done bit.

If Memory Write Done bit set, using CMD_SREG_W8 command, write to SPI Slave Status Register to the clear Memory Write Done bit.

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 or CMD_SREG_W16 or CMD_SREG_R32 command with address 0x04 and data 0x01.
2. Insert 4 TAR/Dummy and 12 wait cycles.
3. Read the Status back and check that all bits should be cleared except RX FIFO Empty/TX FIFO Empty.

11.11.4 MEMORY READ BY SPI MASTER

1. Initiate a transaction from SPI Master with either CMD_MEM_R8 or CMD_MEM_R16 or CMD_MEM_R32 or CMD_MEM_BLK_R command with address 0x0100.
2. Insert 4 TAR/Dummy and 12 wait cycles.
3. Read the 2 byte Status back and check that all bits should be cleared except RX FIFO Empty, Memory Write Done / Memory Read Busy.

Using CMD_RD_SNGL_FIFO8_FSR or CMD_RD_SNGL_FIFO16_FSR or CMD_RD_SNGL_FIFO32_FSR command poll for Memory Read Done bit in the SPI Slave Status Register and then read the data and clear Memory Read Done bit.

1. Initiate a transaction from SPI Master with either CMD_RD_SNGL_FIFO8_FSR or CMD_RD_SNGL_FIFO16_FSR or CMD_RD_SNGL_FIFO32_FSR command.
2. Insert 4 TAR/Dummy.
3. Read status back from SPI slave as long as Memory Read Done bit is not set and after the Memory Read Done bit is set the Data from the SPI slave is valid.
4. Read the Status back and check that all bits should be cleared except TX FIFO Empty/ RX FIFO Empty bit.
5. Using CMD_SREG_W8 command write to SPI Slave Status Register to the clear Memory Read Done bit.

11.11.5 SIMPLE MODE

1. Configuration
   - EC application code needs to program Memory Base Address0 Register and Memory Write Limit0 Register for setting up the write region for SPI Master.
   - EC application code needs to program Memory Base Address1 Register and Memory Read Limit1 Register for setting up the read region for SPI Master.
   - EC application code needs to program 0x0005_0000 in System Configuration Register to enable Simple Mode and activate the block
2. Write from SPI Master
   - Wait for SPI Master to write start the transaction
   - EC can read the RX FIFO Byte Counter Register to know how may bytes have been written to the Memory
   - EC may read the data from Memory Base Address0 Register when it is ready.
3. Read from SPI Master
   - The EC application code sets up response at Memory Base Address1 Register
   - The EC application code then writes to System Configuration Register bit 19 EC Data Available to indicate to the SPI slave that the data is available for sending to SPI Master.
4. Transaction completion
   - The SPI Master terminates the transfer after it has received all the bits.
   - EC Data Available is cleared.
   - TX FIFO is cleared
   - Wait for 2.56micro second for RX FIFO to be empty.
5. Start new transfer by jumping to step 2 and repeating the steps.

Note:	The content of the Simple Mode data transfer need to be interpreted by the application code.

© 2021 Microchip Technology Inc.

12.0 I2C/SMBUS INTERFACE

12.1 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]).

12.2 References

1. I2C_SMB Controller Core with Network Layer Support (SMB2) - 16MHz I2C Baud Clock“, Revision 3.7, Core-Level Architecture Specification, Microchip, date 13 May 2020

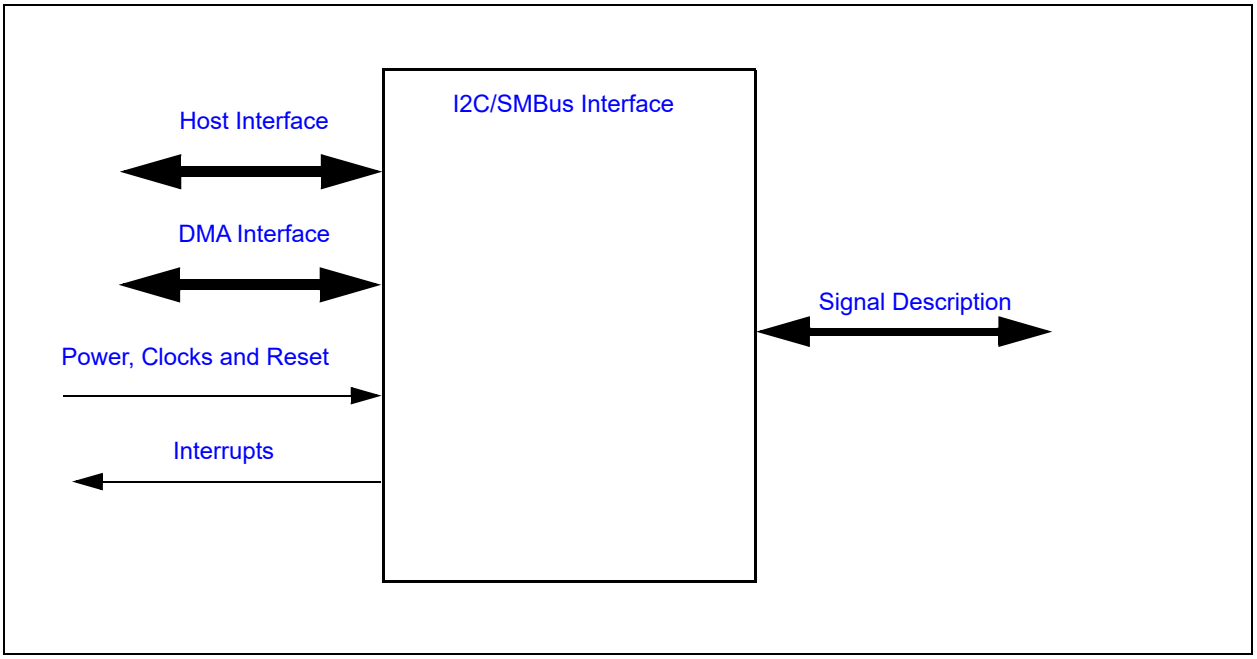
12.3 Terminology

There is no terminology defined for this chapter.

12.4 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 12-1: I/O DIAGRAM OF BLOCK



12.5 Signal Description

see the Pin Configuration section for a description of the SMB-I2C pin configuration.

12.6 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".

12.7 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 8.0, "Internal DMA Controller".

12.8 Power, Clocks and Reset

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

12.8.1 POWER DOMAINS

Name	Description
VTR_CORE	This power well sources all of the registers and logic in this block, except where noted.

12.8.2 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

12.8.3 RESETS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in the SMB-I2C Controller core.

12.9 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.

12.10 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.

© 2021 Microchip Technology Inc.

12.11 Description

12.11.1 SMB-I2C CONTROLLER CORE

The SMB-I2C Controller behavior is defined in the SMB-I2C Controller Core Interface specification (*See* Ref [1]).

12.11.2 PHYSICAL INTERFACE

The Physical Interface for the SMB-I2C Controller core is configurable for up to 106 ports. Each I2C_WAKE Controller can be connected to any of the ports defined in Table 12-1, "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.

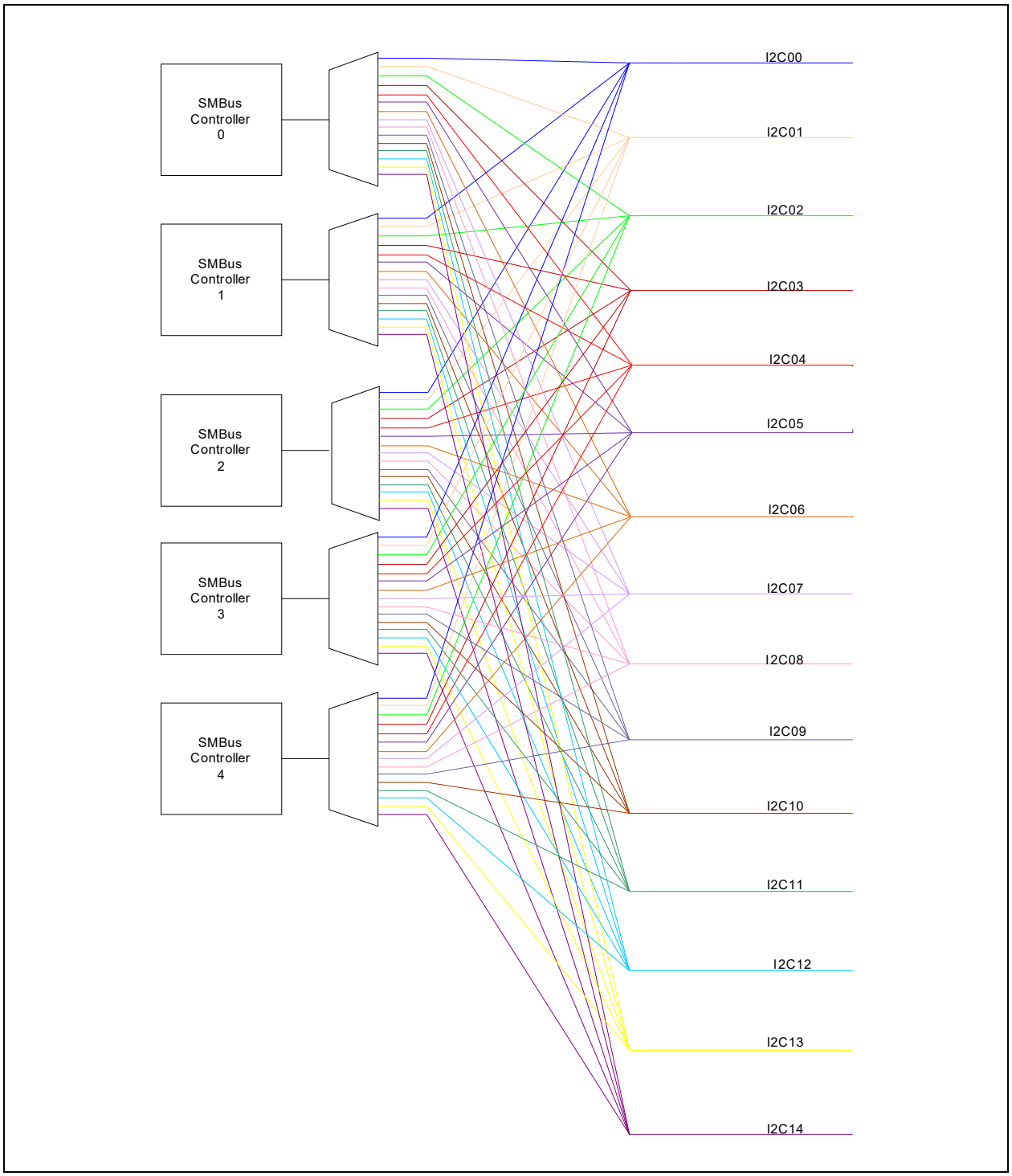
For I^2C port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as an I^2C port.

For more information regarding the SMB-I2C Controller core see *Section 2.2,* "*Physical Interface*" in Ref[1].

TABLE 12-1: 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 2.3for the pin mapping				

FIGURE 12-2: SMB-I2C PORT CONNECTIVITY



© 2021 Microchip Technology Inc.

12.12 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].

12.13 Application Note

Port number and Filter Enable (FEN) should be written before setting the enable bit in the Configuration register. Though a single write can perform the enable as well as configuration simultaneously, it may lead the controller to treat the bus as busy due to noise incurred in configuring the port and Filtering.

For example:

Enable the block after the ports have been set-up.

Config write 0xc0000101 //Set up the port number and Filter enable

Config write 0xc0000501 // Enable the I2C operation

13.0 UART

13.1 Introduction

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

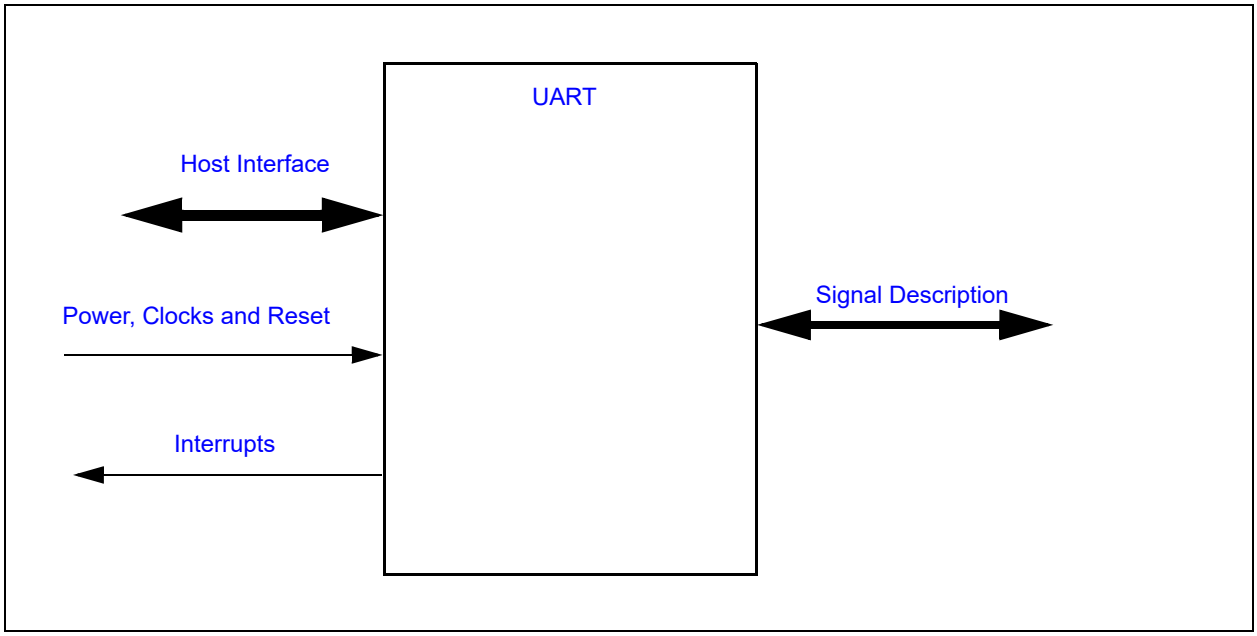
13.2 References

- EIA Standard RS-232-C specification

13.3 Interface

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

FIGURE 13-1: I/O DIAGRAM OF BLOCK



13.4 Signal Description

TABLE 13-1: 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.

© 2021 Microchip Technology Inc.

TABLE 13-1: 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.

13.5 Host Interface

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

13.6 Power, Clocks and Reset

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

13.6.1 POWER DOMAINS

TABLE 13-2: POWER SOURCES

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

13.6.2 CLOCKS

TABLE 13-3: 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 13-4: 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.

13.6.3 RESETS

TABLE 13-5: 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.

13.7 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 13-6: SYSTEM INTERRUPTS

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 13-12, "Interrupt Control Table".

TABLE 13-7: EC INTERRUPTS

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 13-12, "Interrupt Control Table".

13.8 Low Power Modes

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

© 2021 Microchip Technology Inc.

13.9 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.

13.9.1 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 13-8: 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

© 2021 Microchip Technology Inc.

TABLE 13-9: 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

13.10 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 13-10: 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.

© 2021 Microchip Technology Inc.

13.10.1 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

13.10.2 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

13.10.3 PROGRAMMABLE BAUD RATE GENERATOR LSB REGISTER

Offset	00h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	BAUD_RATE_DIVISOR_LSB See Section 13.9.1, "Programmable Baud Rate".	R/W	0h	RESET

13.10.4 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 13.9.1, "Programmable Baud Rate".	R/W	0h	RESET

13.10.5 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 EEC1727. 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

© 2021 Microchip Technology Inc.

13.10.6 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 13-11: RECV FIFO TRIGGER LEVELS

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

13.10.7 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 13-12). 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 13-12, "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

© 2021 Microchip Technology Inc.

TABLE 13-12: 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

13.10.8 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 13-13 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

© 2021 Microchip Technology Inc.

TABLE 13-13: 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 13-14: 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.

13.10.9 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

© 2021 Microchip Technology Inc.

13.10.10 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

© 2021 Microchip Technology Inc.

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

© 2021 Microchip Technology Inc.

13.10.11 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

© 2021 Microchip Technology Inc.

13.10.12 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

13.11 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 13-15: CONFIGURATION REGISTER SUMMARY

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

13.11.1 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

© 2021 Microchip Technology Inc.

13.11.2 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

14.0 GPIO INTERFACE

14.1 General Description

The EEC1727 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.

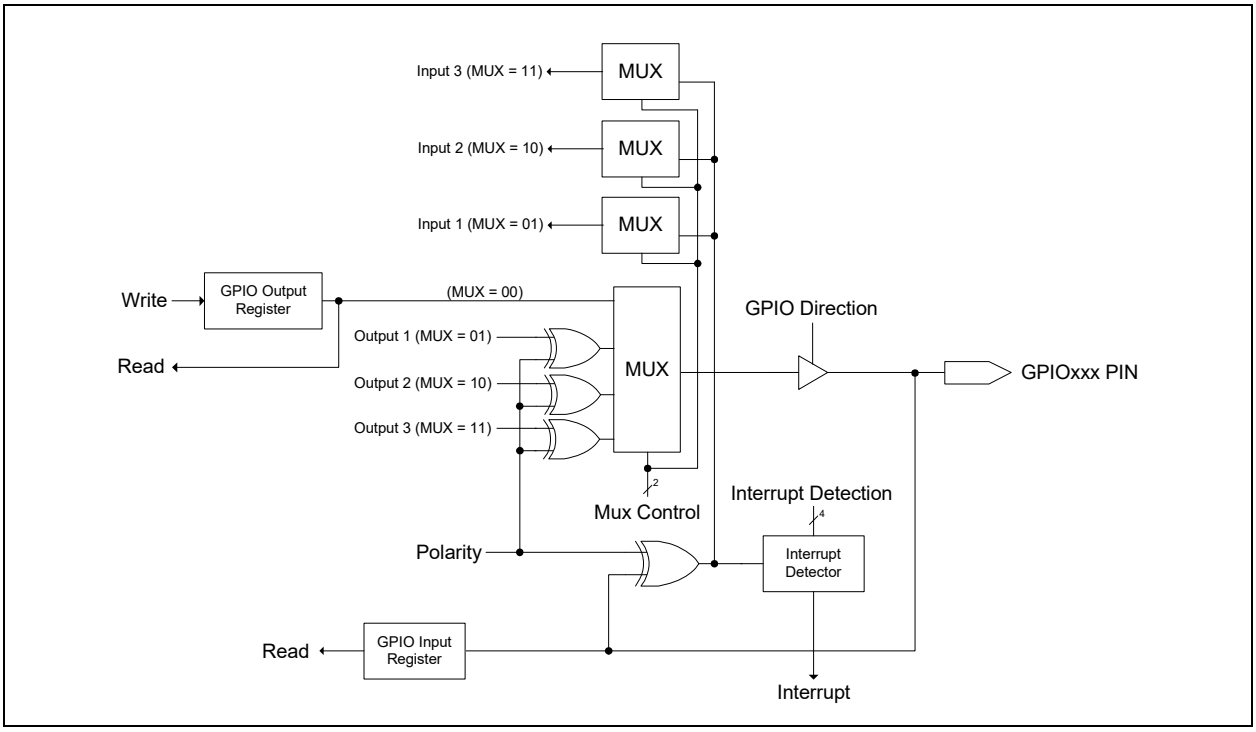
Features of the GPIO Interface include:

- 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

14.2 Block Diagram

The GPIO Interface Block Diagram shown in Figure 14-1 illustrates the functionality of a single EEC1727 GPIO Interface pin. The source for the Pin Multiplexing Control, Interrupt Detection (int_det), GPIO Direction, and Polarity controls in Figure 14-1 is a Pin Control Register that is associated with each pin (see Section 14.7.1.1, "Pin Control Register," on page 217).

FIGURE 14-1: GPIO INTERFACE BLOCK DIAGRAM



© 2021 Microchip Technology Inc.

14.3 Power, Clocks and Reset

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

14.3.1 POWER DOMAINS

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

14.3.2 CLOCK INPUTS

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

14.3.3 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. This reset is used for VCC Power Well Emulation.

14.4 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.

14.5 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 14-1: GPIO Interface Block Diagram on page 212.

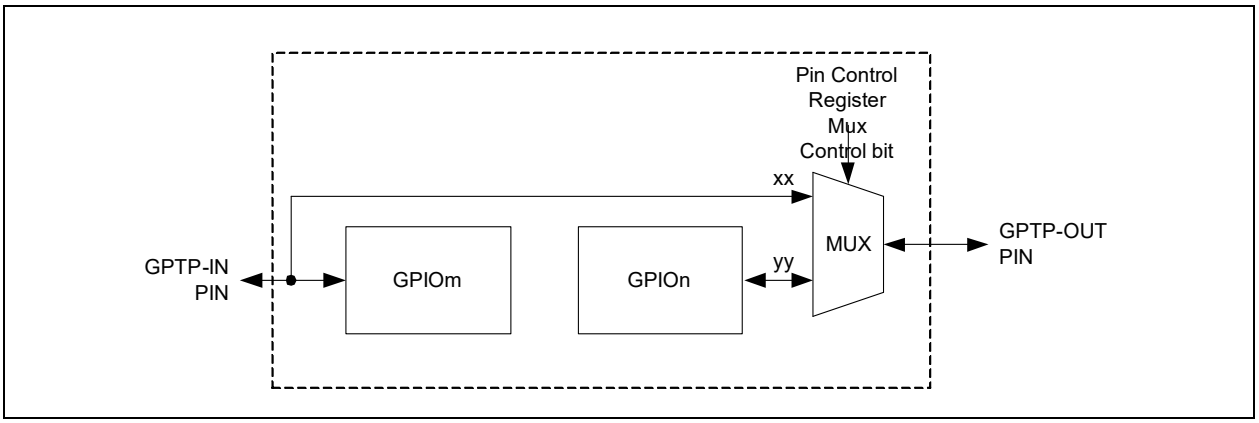
© 2021 Microchip Technology Inc.

14.6 GPIO Pass-Through Ports

GPIO Pass-Through Ports (GPTP) can multiplex two general purpose I/O pins as shown in Figure 14-2. GPIO Pass-Through Ports connect the GPTP-IN pin to the GPTP-OUT pin. The GPTP are sequentially assigned values 0:7. The GPTP port assignment have no relation to the GPIO Indexing assignments. The GPTP ports are controlled by the Mux Control bits in the Pin Control Register associated with the GPTP-OUT signal function.

In order to enable the GPTP Pass-Through Mode, the GPTP-IN (GPIOm in Figure 14-2) Pin Control Register must assign the Mux Control to the GPTP_IN signal function and the GPIO Direction bit to 0 (input); the GPTP-OUT (GPIOn in Figure 14-2) Pin Control Register must assign the Mux Control to the GPTP_OUT signal function and the GPIO Direction bit to 1 (output). The GPTP-OUT signal function can differ from pin to pin.

FIGURE 14-2: GPIO PASS-THROUGH PORT EXAMPLE



The Pin Control Register Mux Control fields shown in Figure 14-2 are illustrated as ‘xx’ and ‘yy’ because this figure is an example, it does not represent the actual GPIO multiplexing configuration. The GPIO Multiplexing tables in this chapter must be used to determine the correct values to use to select between a GPIO and the pass-through.

When Pass-Through Mode is enabled, the GPIOn output is disconnected from the GPIOn pin and the GPIOm pin signal appears on GPIOn pin. Note that in this case the GPIOm input register still reflects the state of the GPIOm pin.

14.6.1 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.

14.6.2 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 EEC1727 GPIO indexing is done sequentially starting from ‘GPIO000.’

© 2021 Microchip Technology Inc.

14.6.3 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.

14.6.3.1 Pin Control Register Defaults

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

14.7 GPIO Registers

The registers listed in the Register Summary table are for a single instance of the EEC1727. 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 14-1: REGISTER BASE ADDRESS TABLE

Instance Name	Instance Number	Host	Address Space	Base Address
GPIO	0	EC	32-bit internal address space	4008_1000h Note 14-1

Note 14-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

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 14-2: 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

© 2021 Microchip Technology Inc.

TABLE 14-2: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
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]
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

© 2021 Microchip Technology Inc.

TABLE 14-2: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
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

14.7.1 PIN CONTROL REGISTERS

Two Pin Control Registers are implemented for each GPIO. The Pin Control Register format is described in Section 14.7.1.1, "Pin Control Register," on page 217. The Pin Control Register 2 format is described in Section 14.7.1.2, "Pin Control Register 2," on page 222. Pin Control Register address offsets and defaults for each product are defined in Section 14.6.3.1, "Pin Control Register Defaults," on page 215.

14.7.1.1 Pin Control Register

Offset	See Table 14-2, "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 14-1	RESET_SYS
23:17	RESERVED	RES	-	-

Offset	See Table 14-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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 14-1	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 14-1	RESET_SYS
14:12	Mux Control The Mux Control field determines the active signal function for a pin. 000 = GPIO Function Selected 001 = Signal Function 1 Selected 010 = Signal Function 2 Selected 011 = Signal Function 3 Selected 100 = Signal Function 4 Selected 101 = Signal Function 5 Selected if applicable for that GPIO, otherwise Reserved 110 = Reserved 111 = Reserved	R/W	Note 14-1	RESET_SYS
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 14-3, "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 14-1	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	See Table 14-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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 14.6.1, "Accessing GPIOs".	R/W	Note 14-1	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 14-1	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 14-1	RESET_SYS
7	Edge Enable (edge_en) 0 = Edge detection disabled 1 = Edge detection enabled **Note:** See Table 14-3, "Edge Enable and Interrupt Detection Bits Definition".	R/W	Note 14-1	RESET_SYS

Offset	See Table 14-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6:4	Interrupt Detection (int_det) The interrupt detection bits determine the event that generates a GPIO_Event. **Note:** See Table 14-3, "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 14-1	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 01 = VCC 10 = Unpowered. The always unpowered setting on a GPIO will force the pin to tri-state. 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 14-1	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 14-1	RESET_SYS

Note 14-1 See Section 3.5, "GPIO Register Assignments" for the default values and Table 14-2, "Register Summary" and Table 3-5, "Register Map" for register offset value for each GPIO Pin Control Register.

Note 14-2 Repeater mode is not available on over voltage protected pins.

© 2021 Microchip Technology Inc.

TABLE 14-3: 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.

14.7.1.2 Pin Control Register 2

Offset	See Note 14-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5:4	DRIVE_STRENGTH These bits are used to select the drive strength on pad type PIO-12. See Note 1. 00 = 2mA 01 = 4mA 10 = 8mA 11 = 12mA These bits are used to select the drive strength on pad type PIO-24. See Note 1. 00 = 4mA 01 = 8mA 10 = 16mA 11 = 24mA	R/W	00	RESET_SYS
3:1	Reserved	RES	-	-
0	SLEW_RATE This bit is used to select the slew rate on the pin. 1=fast 0=slow (half frequency)	R/W	0h	RESET_SYS
Note 1: The drive strength is dependent on the Pad type define in Table 2-2, "EEC1727 68 WFBGA PIN MUX TABLE" of the Pin Chapter. See Electrical Specification Table 37-3, "DC Electrical Characteristics" for drive strength options per pad.				

14.7.2 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.

© 2021 Microchip Technology Inc.

14.7.2.1 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

14.7.2.2 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

14.7.2.3 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

© 2021 Microchip Technology Inc.

Offset	388h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:8	GPIO[117:110] Output	R/W	00h	RESET_SYS
7:0	GPIO[107:100] Output	R/W	00h	RESET_SYS

14.7.2.4 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
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

14.7.2.5 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

© 2021 Microchip Technology Inc.

14.7.2.6 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
15:8	GPIO[257:250] Output	R/W	00h	RESET_SYS
7:0	GPIO[247:240] Output	R/W	00h	RESET_SYS

14.7.3 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.

14.7.3.1 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

14.7.3.2 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

14.7.3.3 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

© 2021 Microchip Technology Inc.

14.7.3.4 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

14.7.3.5 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

14.7.3.6 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

© 2021 Microchip Technology Inc.

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

© 2021 Microchip Technology Inc.

15.0 WATCHDOG TIMER (WDT)

15.1 Introduction

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

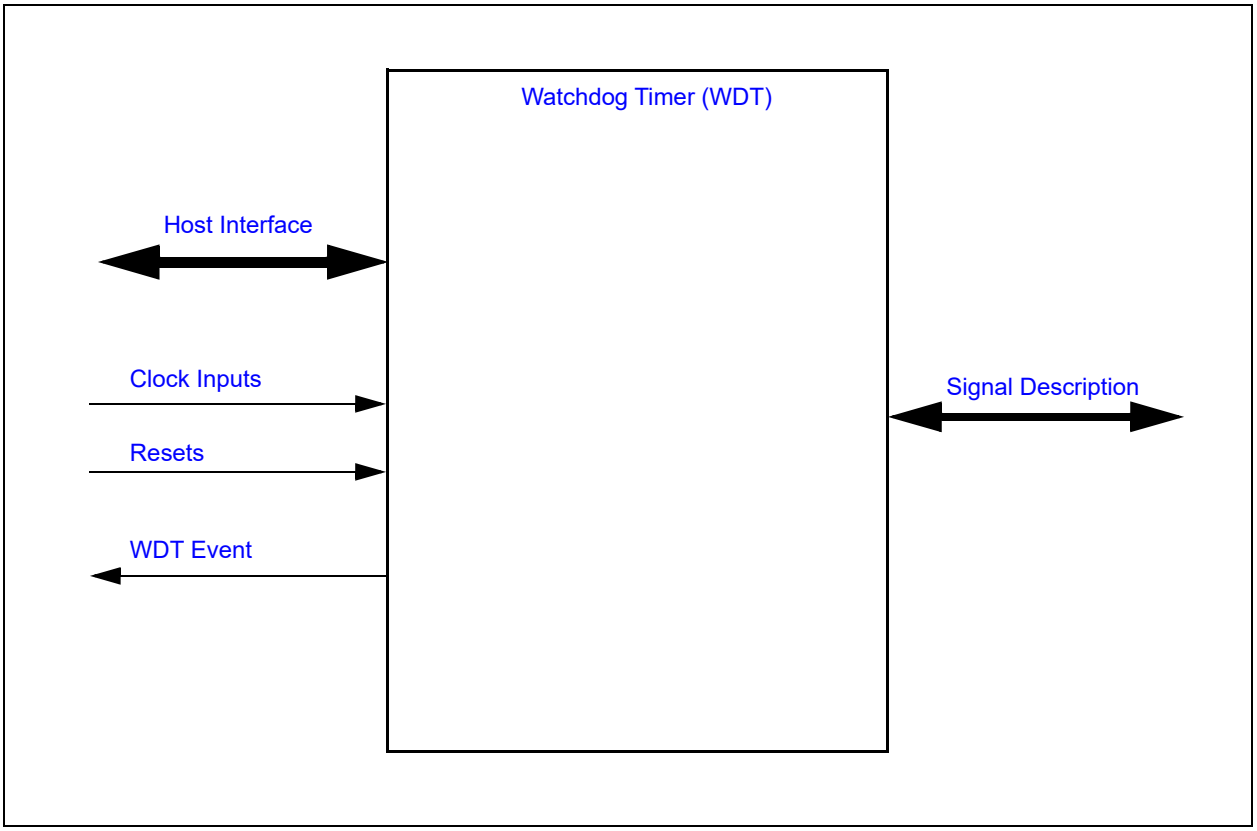
When enabled, the Watchdog Timer (WDT) circuit will generate a WDT Event if the user program fails to reload the WDT within a specified length of time known as the WDT Interval.

15.2 Interface

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

15.3 Host Interface

FIGURE 15-1: I/O DIAGRAM OF BLOCK



The registers defined for the Watchdog Timer (WDT) are accessible by the embedded controller as indicated in Section 15.7, "EC Registers". All registers accesses are synchronized to the host clock and complete immediately. Register reads/writes are not delayed by the 32KHz Core.

15.4 Signal Description

15.4.1 SIGNAL INTERFACE

There are no external signals for this block.

15.5 Power, Clocks and Reset

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

15.5.1 POWER DOMAINS

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

15.5.2 CLOCK INPUTS

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

15.5.3 RESETS

TABLE 15-1: 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 15-2: 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.

15.6 Description

15.6.1 WDT OPERATION

15.6.1.1 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.

15.6.1.2 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.

© 2021 Microchip Technology Inc.

15.6.1.3 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 479 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.

15.6.1.4 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 Core seconds (ex. 33/32.768 KHz = 1.007ms) to decrement by 1 count.

15.6.1.5 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.

15.7 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 15-3: 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

15.7.1 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

15.7.2 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 subsystem 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 15.6.1.1, "WDT Activation Mechanism" and deactivated by the sequence of operations defined in Section 15.6.1.2, "WDT Deactivation Mechanism". 1=block enabled 0=block disabled	R/W	0b	RESET_SYS

© 2021 Microchip Technology Inc.

15.7.3 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

15.7.4 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

15.7.5 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

15.7.6 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

© 2021 Microchip Technology Inc.

16.0 16/32 BIT BASIC TIMER

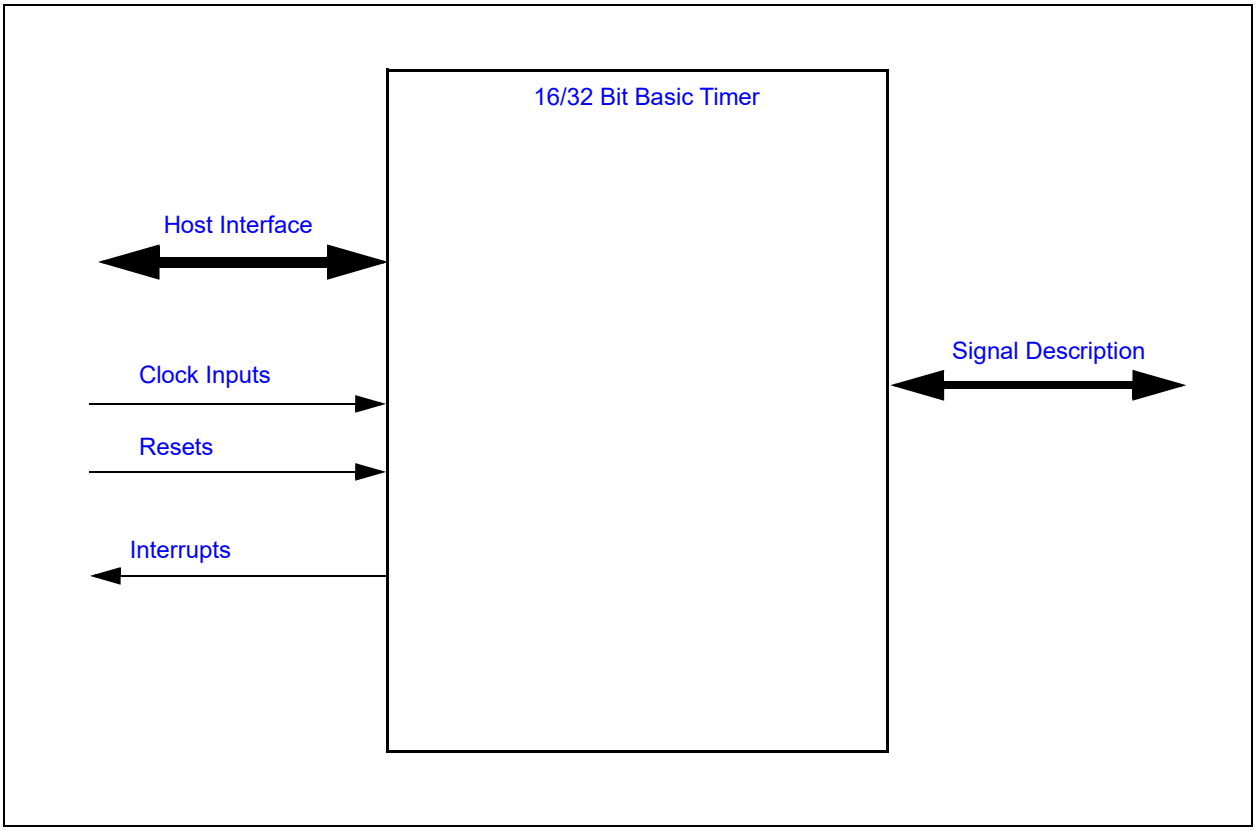
16.1 Introduction

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.

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

There are no external signals for this block.

16.4 Host Interface

The Embedded Controller (EC) may access this block via the registers defined in Section 16.9, "EC-Only Registers," on page 237.

16.5 Power, Clocks and Reset

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

16.5.1 POWER DOMAINS

TABLE 16-1: POWER SOURCES

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

16.5.2 CLOCK INPUTS

TABLE 16-2: 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.

16.5.3 RESETS

TABLE 16-3: 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.

16.6 Interrupts

TABLE 16-4: 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.	

© 2021 Microchip Technology Inc.

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.

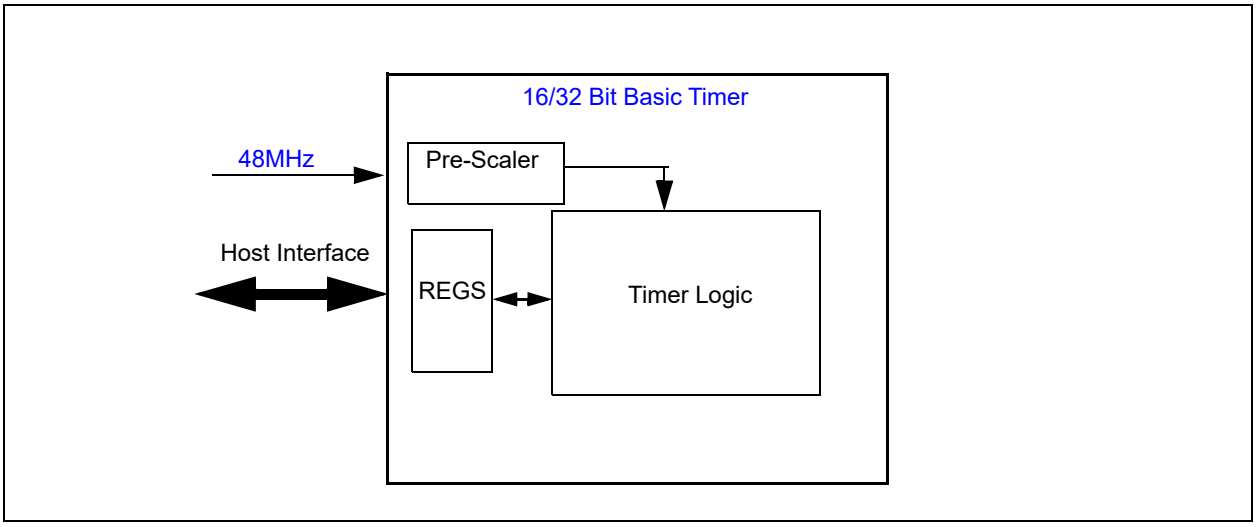
The sleep state of this timer is as follows:

- 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.

16.8 Description

FIGURE 16-2: 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.

16.9 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 16-5: "EEC1727 Instance Naming Convention"**.

TABLE 16-5: EEC1727 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 16-6: 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

16.9.1 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

© 2021 Microchip Technology Inc.

16.9.2 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

16.9.3 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

16.9.4 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

16.9.5 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

© 2021 Microchip Technology Inc.

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

17.0 16-BIT COUNTER-TIMER INTERFACE

17.1 Introduction

The 16-Bit Counter-Timer Interface implements four 16-bit auto-reloading timer/counters. The clock for each timer/counter is derived from the system clock and can be divided down by a prescaler. Input-Only and Input/Output timers can also use an external input pin to clock or gate the counter. To aid operation in noisy environments the external input pin also has a selectable noise filter. If large counts are required, the output of each timer/counter can be internally connected to the next timer/counter.

17.2 References

No references have been cited for this feature.

17.3 Terminology

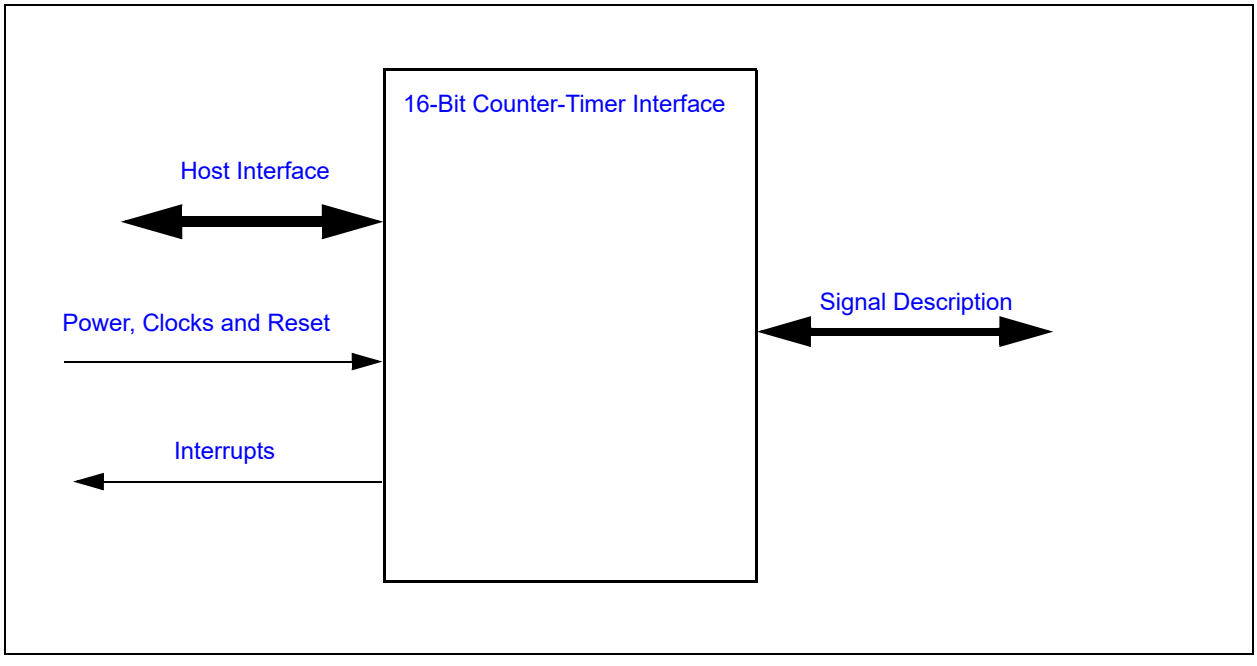
TABLE 17-1: TERMINOLOGY

Term	Definition
Overflow	When the timer counter transitions from FFFFh to 0000h.
Underflow	When the timer counter transitions from 0000h to FFFFh.
Timer Tick Rate	This is the rate at which the timer is incremented or decremented.

17.4 Interface

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

FIGURE 17-1: I/O DIAGRAM OF BLOCK



© 2021 Microchip Technology Inc.

17.5 Signal Description

TABLE 17-2: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
TINx	INPUT	Timer x Input signal
TOUTx	OUTPUT	Timer x Output signal

17.6 Host Interface

The registers defined for 16-bit Timers are accessible by the various hosts as indicated in Section 17.11, "EC Registers".

17.7 Power, Clocks and Reset

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

17.7.1 POWER DOMAINS

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

17.7.2 CLOCK INPUTS

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

17.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their 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 RESET bit is set in the Timer x Control Register.
Reset_Timer	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.

17.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
TIMER*x*	This interrupt event fires when a 16-bit timer *x* overflows or underflows.

17.9 Low Power Modes

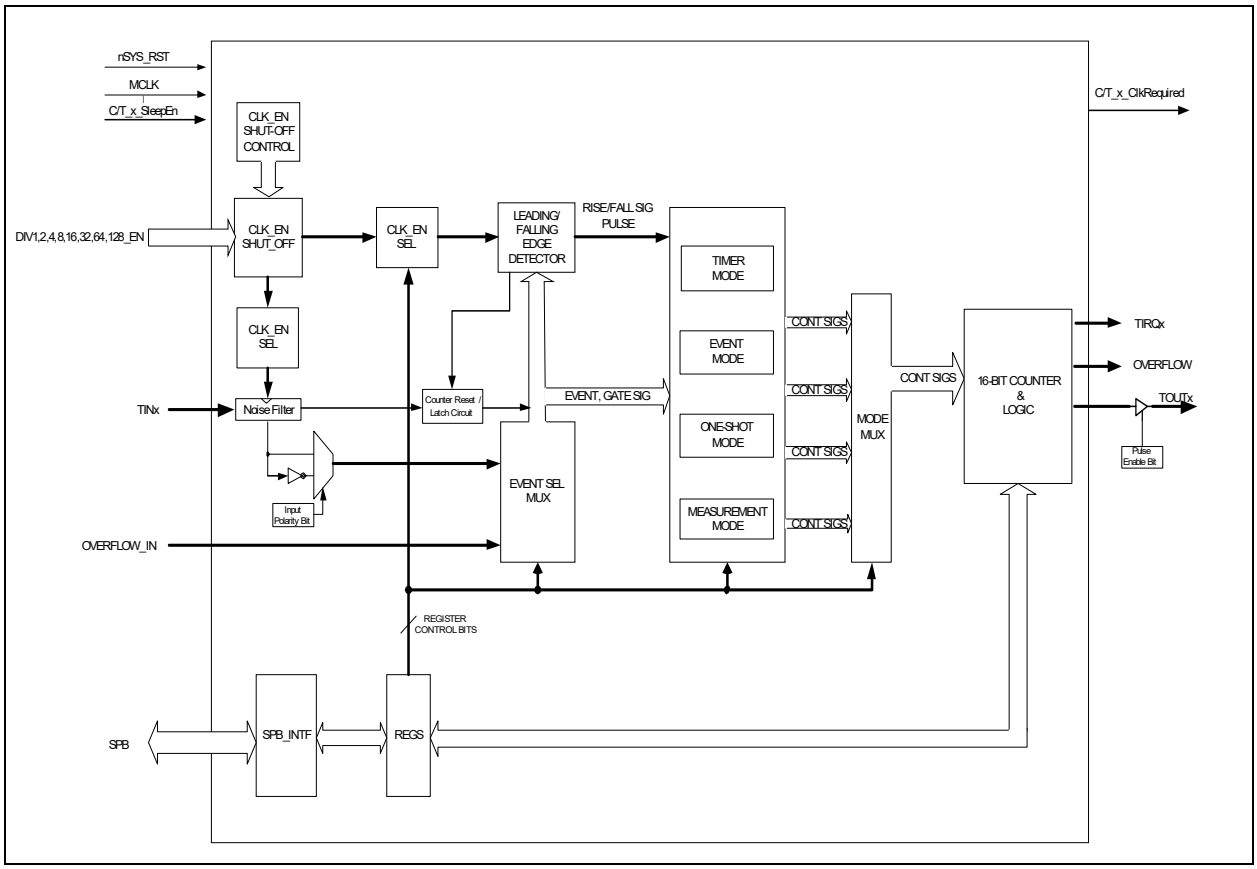
The 16-bit Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active. The block is inactive in the following conditions:

- The block is not running (ENABLE de-asserted)
- The block is powered down (PD asserted).

The timer requires one Timer Clock period to halt after receiving a Sleep_En signal. When the block returns from sleep, if enabled, it will be restarted from the preload value.

17.10 Description

FIGURE 17-2: BLOCK DIAGRAM FOR TIMER X



The 16-bit Timer consists of a 16-bit counter, clocked by a by a configurable Timer Clock. The Timer can operate in any of 4 Modes: Timer Mode, Event Mode, One-Shot Mode, and Measurement Mode. The Timer can be used to generate an interrupt to the EC. Depending on the mode, the Timer can also generate an output signal.

17.10.1 TIMER CLOCK

Any of the frequencies listed in Table 17-3 may be used as the time base for the 16-bit counter.

TABLE 17-3: 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

© 2021 Microchip Technology Inc.

TABLE 17-3: TIMER CLOCK FREQUENCIES (CONTINUED)

Timer Clock Select	Frequency Divide Select	Frequency Selected
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 Timer x Clock and Event Control Register.

17.10.2 FILTER CLOCK AND NOISE FILTER

The noise filter uses the Filter Clock (FCLK) to filter the signal on the TINx pins. for Event Mode and One-Shot Mode.

In Event Mode, the Event input is synchronized to FCLK and (if enabled) filtered by a three stage filter. The resulting recreated clock is used to clock the timer in Event mode. In Bypass Mode, configured by the FILTER_BYPASS bit in the Timer x Control Register, the pulse width of the external signal must be at least 2x the pulse width of the FCLK source. In Filter Mode, the pulse width of the external signal must be at least 4x the pulse width of the sync and filter clock.

In One-Shot mode, the TIN duration could be smaller than a TCLK period. The filtered signal is latched until the signal is seen in the TCLK domain. This also applies in the filter bypass mode

Frequencies for the Filter Clock are the as those available for the Timer Clock, and are listed in Table 17-3. For the Filter Clock, the **Timer Clock Select** value is defined by the FCLK field in the Timer x Clock and Event Control Register. The choice of frequency is independent of the value chosen for the Timer Clock.

17.10.3 TIMER CONNECTIONS

For external inputs/outputs (TINx/TOUTx) to/from timers, please see Pin Configuration chapter for a description of the 16-bit Counter/Timer Interface.

TABLE 17-4: TIMER CASCADING DESCRIPTION

Timer Name	Timer Type	Over-Flow/ Under-flow Input's Connection
Timer 0	General Purpose	from Timer 3
Timer 1	General Purpose	from Timer 0
Timer 2	General Purpose	from Timer 1
Timer 3	General Purpose	from Timer 2

Note: The cascading connections are independent of the TINx/TOUTx connections.

17.10.4 STARTING AND STOPPING

The 16-bit timers can be started and stopped by setting and clearing the ENABLE bit in the Timer x Control Register in all modes, except one-shot.

17.10.5 TIMER MODE

Timer mode is used to generate periodic interrupts to the EC. When operating in this mode the timer always counts down based on one of the internally generated clock sources. The Timer mode is selected by setting the Timer Mode Select bits in the Timer Control Register. See Section 17.11.1, "Timer x Control Register".

The period between timer interrupts and the width of the output pulse is determined by the speed of the clock source, the clock divide ratio and the value programmed into the Timer Reload Register. The timer clock source and clock rate are selected using the Clock Source Select bits (TCLK) in the Timer x Clock and Event Control Register. See Section 17.11.2, "Timer x Clock and Event Control Register".

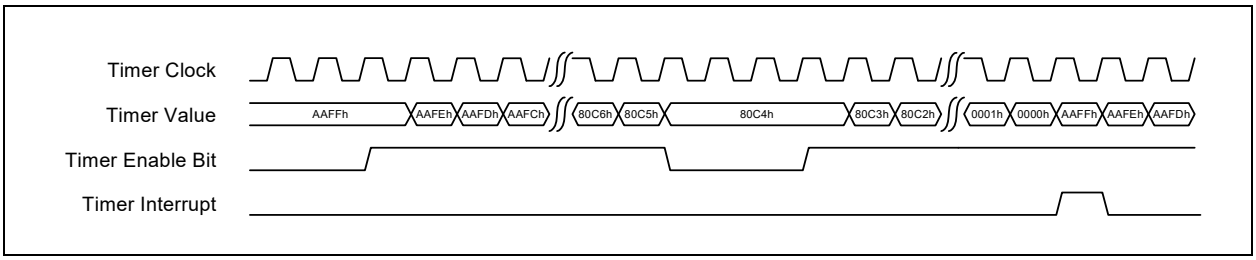
TABLE 17-5: TIMER MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Count Operation	Down Counter
Reload Operation	When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh.
Count Start Condition	UPDN = 0 (timer only mode): ENABLE = 1 UPDN = 1 (timer gate mode): ENABLE = 1 & TIN = 1;
Count Stop Condition	UPDN = 0=ENABLE = 0; UPDN = 1: (ENABLE= 0 | TIN = 0)
Interrupt Request Generation Timing	When timer underflows from 0000h to reload value (as determined by RLOAD) an interrupt is generated.
TINx Pin Function	Provides timer gate function
TOUTx Pin Function	TOUT toggles each time the timer underflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows. When the timer is running, values written to the Timer Reload Register are written to the timer counter when the timer underflows. The assertion of Reset also copies the Timer Reload Register into the timer counter.
Selectable Functions	• Reload timer on underflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Timer can be started and stopped by the TINx input pin (Gate Function) • The TOUTx pin changes polarity each time the timer underflows (Pulse Output Function)

17.10.5.1 Timer Mode Underflow

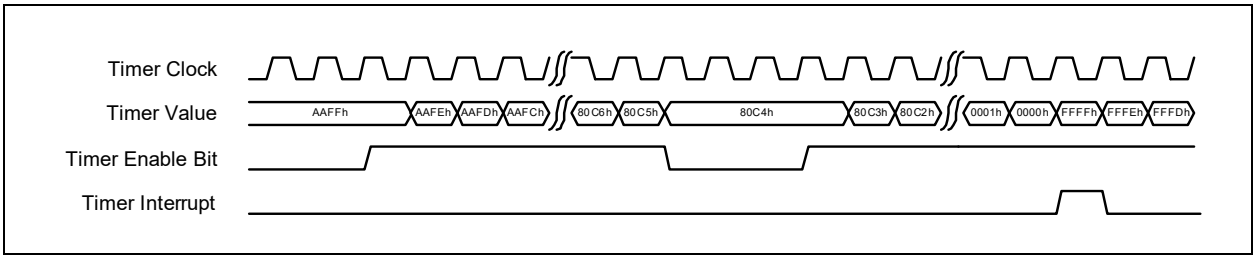
The timer operating in Timer mode can underflow in two different ways. One method, the Reload mode shown in Figure 17-3, is to reload the value programmed into the Reload register and continue counting from this value. The second method, Free Running mode Figure 17-4, is to set the timer to FFFFh and continue counting from this value. The underflow behavior is controlled by the RLOAD bit in the Timer Control Register.

FIGURE 17-3: RELOAD MODE BEHAVIOR



© 2021 Microchip Technology Inc.

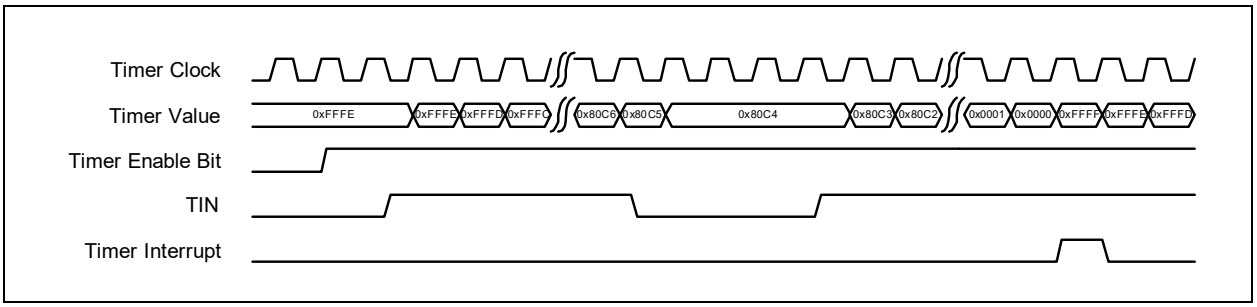
FIGURE 17-4: FREE RUNNING MODE BEHAVIOR



17.10.5.2 Timer Gate Function

The TIN pin on each timer can be used to pause the timer's operation when the timer is running. The timer will stop counting when the TIN pin is deasserted and count when the TIN pin is asserted. Figure 17-5 shows the timer behavior when the TIN pin is used to gate the timer function. The UPDN bit is used to enable and disable the Timer Gate function when in the Timer mode.

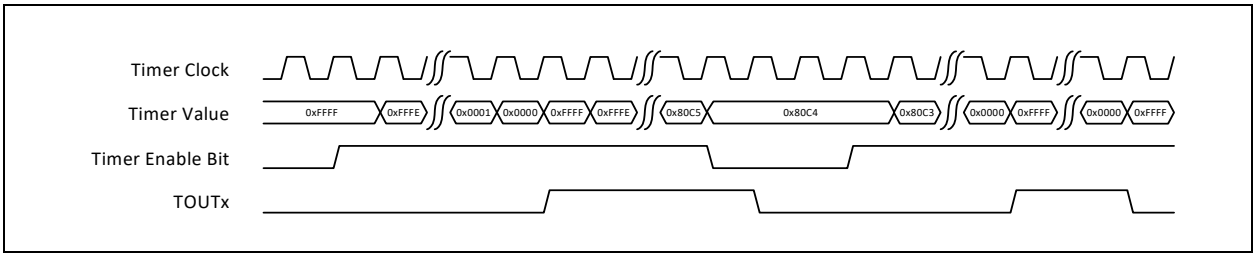
FIGURE 17-5: TIMER GATE OPERATION



17.10.5.3 Timer Mode Pulse Output

The four Timers can be used to generate a periodic output pulse. The output pulse changes state each time the timer underflows. The output is also cleared when the EN bit is cleared. Figure 17-6 shows the behavior of the TOUTx pin when it is used as a pulse output pin.

FIGURE 17-6: TIMER PULSE OUTPUT



17.10.6 EVENT MODE

Event mode is used to count events that occur external to the timer. The timer can be programmed to count the overflow output from the previous timer or an edge on the TIN pin. The direction the timer counts in Event mode is controlled by the UPDN bit in the Timer Control Register. When the timer is in Event mode, the TOUTx signal can be used to generate a periodic output pulse when the timer overflows or underflows. Figure 17-6 illustrates the pulse output behavior of the TOUTx pin in event mode when the timer underflows.

© 2021 Microchip Technology Inc.

The timer can be programmed using the Clock and Event Control register to respond to the following events using the EVENT bits and the EDGE bits: rising edge of TINx, falling edge of TINx, rising and falling edge of TINx, rising edge of overflow input, falling edge of the overflow input, and the rising and falling edges of the overflow input.

TABLE 17-6: EVENT MODE OPERATIONAL SUMMARY

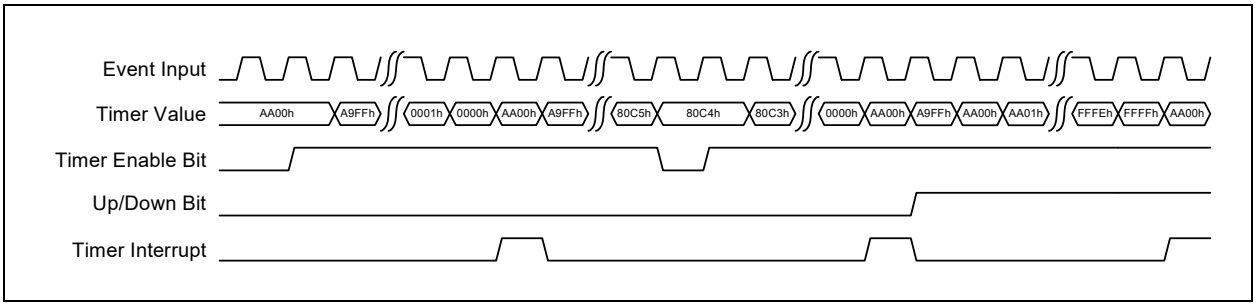
Item	Description
Count Source	• External signal input to TINx pin (effective edge can be selected by software) • Timer x-1 overflow
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Count Operation	Up/Down Counter
Reload Operation	• When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh. • When the timer overflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to 0000h.
Count Start Condition	Timer Enable is set (ENABLE = 1)
Count Stop Condition	Timer Enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	When timer overflows or underflows
TINx Pin Function	Event Generation
TOUTx Pin Function	TOUT toggles each time the timer underflows/overflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• The direction of the counter is selectable via the UPDN bit. • Reload timer on underflow/overflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Pulse Output Function The TOUTx pin changes polarity each time the timer underflows or overflows.

17.10.6.1 Event Mode Operation

The timer starts counting events when the ENABLE bit in the Timer Control Register is set and continues to count until the ENABLE bit is cleared. When the ENABLE bit is set, the timer continues counting from the current value in the timer except after a reset event. After a reset event, the timer always starts counting from the value programmed in the Reload Register if counting down or from 0000h if counting up. Figure 17-7 shows an example of timer operation in Event mode. The RLOAD bit controls the behavior of the timer when it underflows or overflows.

© 2021 Microchip Technology Inc.

FIGURE 17-7: EVENT MODE OPERATION



17.10.7 ONE-SHOT MODE

The One-Shot mode of the timer is used to generate a single interrupt to the EC after a specified amount of time. The timer can be configured to start using the ENABLE bit (Figure 17-8) or on a timer overflow event from the previous timer. See Section 17.11.2, "Timer x Clock and Event Control Register" for configuration details. The ENABLE bit must be set for an event to start the timer. The ENABLE bit is cleared one clock after the timer starts. The timer always starts from the value in the Reload Register and counts down in One-Shot mode.

TABLE 17-7: ONE SHOT MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Count Operation	Down Counter
Reload Operation	When the timer underflows the timer will stop. When the timer is enabled timer starts counting from value programmed in Timer Reload Register. (RLOAD has no effect in this mode)
Count Start Condition	Setting the ENABLE bit to 1 starts One-Shot mode. The timer clock automatically clears the enable bit one timer tick later. One-Shot mode may be enabled in Event Mode. In Event mode an overflow from the previous timer is used for timer tick rate.
Count Stop Condition	• Timer is reset (RESET = 1) • Timer underflows
Interrupt Request Generation Timing	When an underflow occurs.
TINx Pin Function	One Shot External input
TOUTx Pin Function	The TOUTx pin is asserted when the timer starts and de-asserted when the timer stops
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• Pulse Output Function The TOUTx pin is asserted when the timer starts and de-asserted when the timer stops.

FIGURE 17-8: TIMER START BASED ON ENABLE BIT

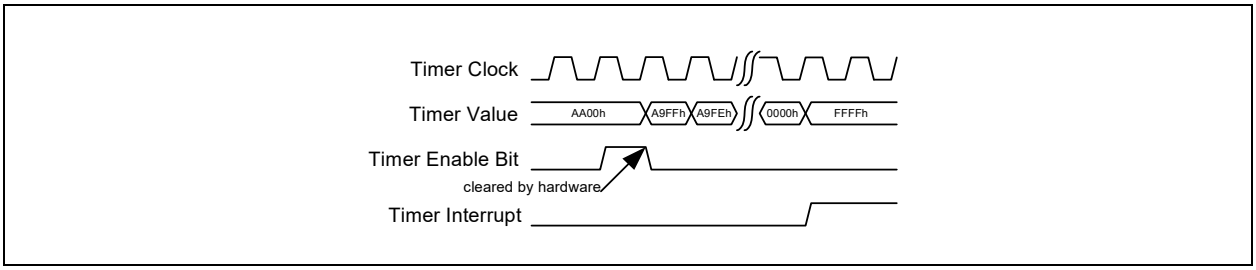


FIGURE 17-9: TIMER START BASED ON EXTERNAL EVENT

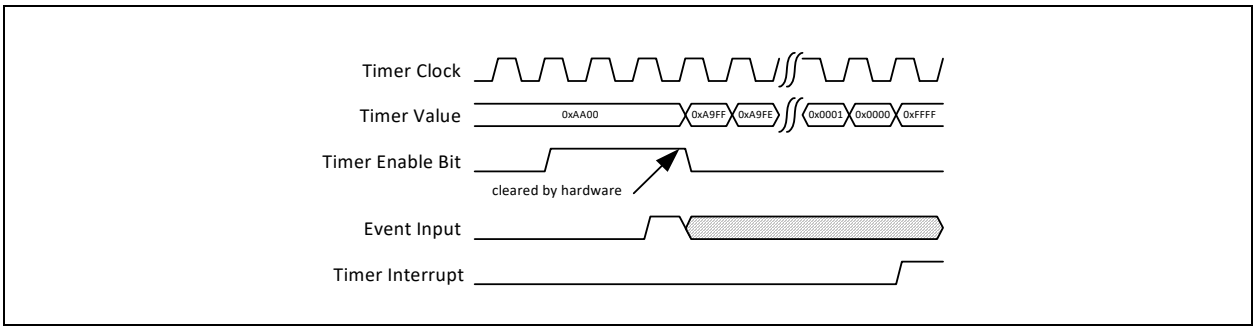
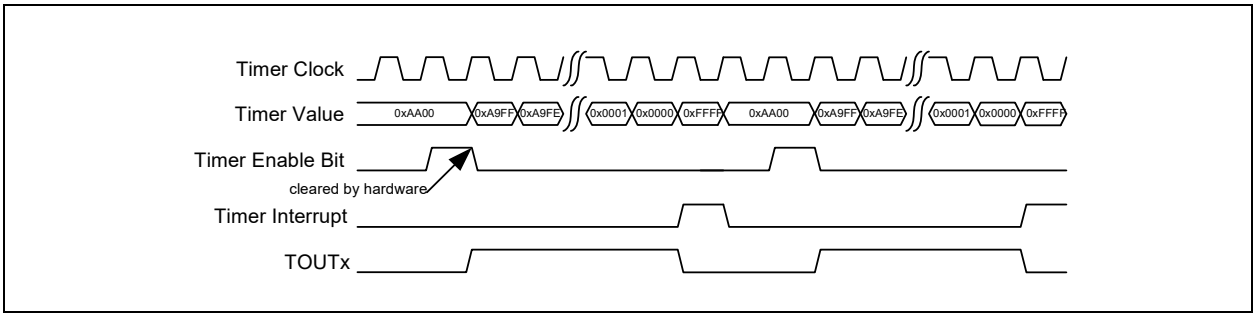


FIGURE 17-10: ONE SHOT TIMER WITH PULSE OUTPUT



17.10.8 MEASUREMENT MODE

The Measurement mode is used to measure the pulse width or period of an external signal. An interrupt to the EC is generated after each measurement or if the timer overflows and no measurement occurred. The timer measures the pulse width or period by counting the number of clock between edges on the TINx pin. The timer always stars counting at zero and counts up to 0xFFFF. The accuracy of the measurement depends on the speed of the clock being used. The speed of the clock also determines the maximum pulse width or period that can be detected.

TABLE 17-8: MEASUREMENT MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 17-3, "Timer Clock Frequencies"

© 2021 Microchip Technology Inc.

TABLE 17-8: MEASUREMENT MODE OPERATIONAL SUMMARY (CONTINUED)

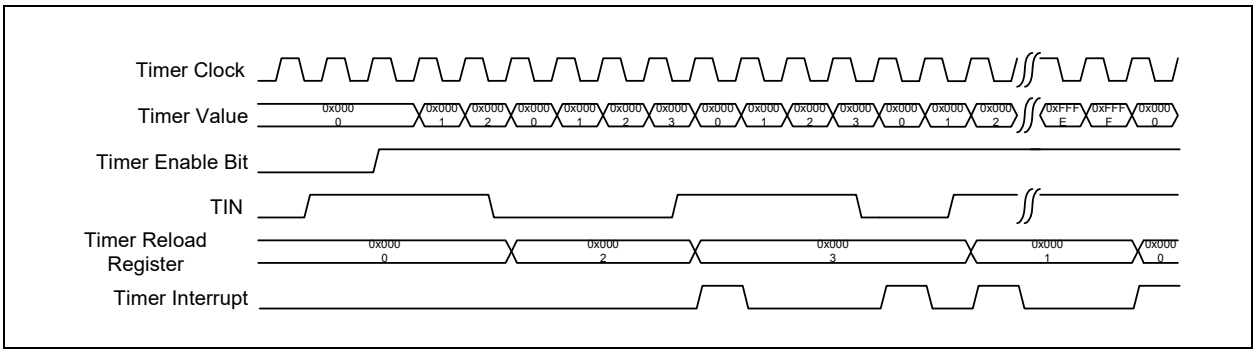
Item	Description
Count Operation	• Up Count • At measurement pulse's effective edge, the count value is transferred to the Timer Reload Register and the timer is loaded with 0000h and continues counting.
Count Start Condition	• Timer enable is set (ENABLE = 1)
Count Stop Condition	• Timer is reset (RESET = 1) • Timer overflows • Timer enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	• When timer overflows • When a measurement pulse's effective edge is input. (An interrupt is not generated on the first effective edge after the timer is started.)
TINx Pin Function	Programmable Input port or Measurement input
Read From Timer	When the Timer x Reload Register is read it indicates the measurement result from the last measurement made. The Timer x Reload Register reads 0000h if the timer overflows before a measurement is made.
Write to Timer	Timer x Reload Register is Read-Only in Measurement mode

17.10.8.1 Pulse Width Measurements

The timers measure pulse width by counting the number of timer clocks since the last rising or falling edge of the TINx input. To measure the pulse width of a signal on the TINx pin, the EDGE bits in the Clock and Event Control Register, must be set to start counting on rising and falling edges. The timer starts measuring on the next edge (rising or falling) on the TINx pin after the ENABLE bit is set. The Reload register stores the result of the last measurement taken. If the timer overflows, 0x0000 is written to the Reload register and the ENABLE bit is cleared stopping the timer. Figure 17-11 shows the timer behavior when measuring pulse widths.

The timer will not assert an interrupt in Pulse Measurement mode until the timer detects both a rising and a falling edge.

FIGURE 17-11: PULSE WIDTH MEASUREMENT

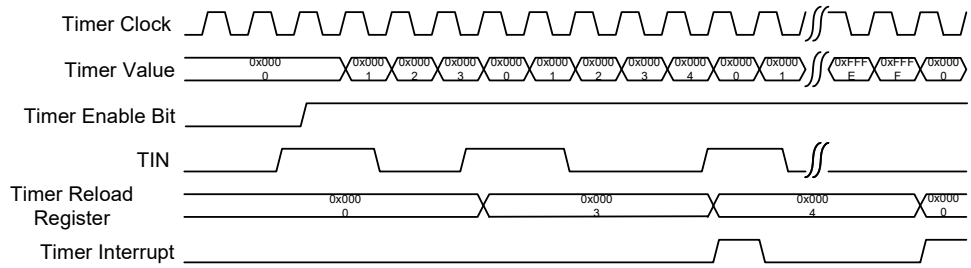


17.10.8.2 Period Measurements

The 16-bit timer measures the period of a signal by counting the number of timer clocks between either rising or falling edges of the TINx input. The measurement edge is determined by the EDGE bits in the Clock and Event Control Register. The timer starts measuring on the next edge (rising or falling) on the TINx pin after the ENABLE bit is set. The reload register stores the result of the last measurement taken. If the timer overflows, 0x0000 is written to the reload register. Figure 17-12 shows the timer behavior when measuring the period of a signal.

The timer will not signal an interrupt in period measurement mode until the timer detects either two rising edges or two falling edges.

FIGURE 17-12: PULSE PERIOD MEASUREMENT



17.11 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 16-Bit Counter-Timer Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 17-9: REGISTER SUMMARY

Offset	Register Name
00h	Timer x Control Register
04h	Timer x Clock and Event Control Register
08h	Timer x Reload Register
0Ch	Timer x Count Register

17.11.1 TIMER X CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:13	Reserved	R	-	-
12	TIMERX_CLK_REQ This bit reflects the current state of the timer's Clock_Required output signal. 1=The main clock is required by this block 0=The main clock is not required by this block	R	0h	Reset_Timer
11	SLEEP_ENABLE This bit reflects the current state of the timer's Sleep_Enable input signal. 1=Normal operation 0=Sleep Mode is requested	R	0h	Reset_Timer

© 2021 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TOUT_POLARITY This bit determines the polarity of the TOUTx output signal. In timer modes that toggle the TOUTx signal, this polarity bit will not have a perceivable difference, except to determine the inactive state. In One-Shot mode this determines if the pulsed output is active high or active low. 1=Active low 0=Active high	R/W	0h	Reset_ Timer
9	PD Power Down. 1=The timer is powered down and all clocks are gated 0=The timer is in a running state	R/W	1h	Reset_ Timer
8	FILTER_BYPASS This bit is used to enable or disable the noise filter on the TINx input signal. 1=IBypass Mode: input filter disabled. The TINx input directly affects the timer 0=Filter Mode: input filter enabled. The TINx input is filtered by the input filter	R/W	0h	Reset_ Timer
7	RLOAD Reload Control. This bit controls how the timer is reloaded on over-flow or underflow in Event and Timer modes. It has no effect in One Shot mode. 1=Reload timer from Timer Reload Register and continue counting 0=Roll timer over to FFFFh and continue counting when counting down and rolls over to 0000h and continues counting when counting up	R/W	0h	Reset_ Timer
6	TOUT_EN This bit enables the TOUTx pin 1=TOUTx pin function is enabled 0=TOUTx pin is inactive	R/W	0h	Reset_ Timer
4	UPDN In Event Mode, this bit selects the timer count direction. In Timer Mode enables timer control by the TINx input pin. Event Mode: 1=The timer counts up 0=The timer counts down Timer Mode: 1=TINx pin pauses the timer when de-asserted 0=TINx pin has no effect on the timer	R/W	0h	Reset_ Timer

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	INPOL This bit selects the polarity of the TINx input 1=TINx is active low 0=TINx is active high	R/W	0h	Reset_ Timer
3:2	MODE Timer Mode. 3=Measurement Mode 2=One Shot Mode 1=Event Mode 0=Timer Mode	R/W	0h	Reset_ Timer
1	RESET This bit stops the timer and resets the internal counter to the value in the Timer Reload Register. This bit also clears the ENABLE bit if it is set. This bit is self-clearing after the timer is reset. Firmware must poll the RESET bit in order to determine when the timer is active after reset. The polling time may be any value from 0 ms to 2^(TCLK+1))/48MHz. If it the TCLK value was set to 0111b then the polling time will be a 5.33us (typ). Worst case polling time is dependent on accuracy of 48MHz clock source. Interrupts are blocked only when RESET takes effect and the ENABLE bit is cleared. If interrupts are not desired, firmware must mask the interrupt in the interrupt block. 1=Timer reset 0=Normal timer operation	R/W	0h	Reset_ Timer
0	ENABLE This bit is used to start and stop the timer. This bit does not reset the timer count but does reset the timer pulse output. This bit will be cleared when the timer stops counting in One-Shot mode. The ENABLE bit is cleared after a RESET cycle has completed. Firmware must poll the RESET bit in order to determine when the timer is active after reset. 1=Timer is enabled 0=Timer is disabled	R/W	0h	Reset_ Timer

© 2021 Microchip Technology Inc.

17.11.2 TIMER X CLOCK AND EVENT CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:12	Reserved	R	-	-
11:8	FCLK Timer Clock Select. This field determines the clock source for the TINx noise filter. See Section 17.10.2, "Filter Clock and Noise Filter" for a description of the available frequencies. The available frequencies are the same as for TCLK.	R/W	0h	Reset_Timer
7	EVENT Event Select. This bit is used to select the count source when the timer is operating in Event Mode. 1=TINx is count source 0=Timer x-1 overflow is count source	R/W	0h	Reset_Timer
6:5	EDGE This field selects which edge of the TINx input signal affects the timer in Event Mode, One-Shot Mode and Measurement Mode. Event Mode: 11b=No event selected 10b=Counts rising and falling edges 01b=Counts rising edges 00b=Counts falling edges One-Shot Mode: 11b=Start counting when the Enable bit is set 10b=Starts counting on a rising or falling edge 01b=Starts counting on a rising edge 00b=Starts counting on a falling edge Measurement Mode: 11b=No event selected 10b=Measures the time between rising edges and falling edges and the time between falling edges and rising edges 01b=Measures the time between rising edges 00b=Measures the time between falling edges	R/W	0h	Reset_Timer
4	Reserved	R	-	-
3:0	TCLK Timer Clock Select. This field determines the clock source for the 16-bit counter in the timer. See Section 17.10.1, "Timer Clock" for a description of the available frequencies.	R/W	0h	Reset_Timer

17.11.3 TIMER X RELOAD REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	TIMER_RELOAD The Timer Reload register is used in Timer and One-Shot modes to set the lower limit of the timer. In Event mode the Timer Reload register sets either the upper or lower limit of the timer depending on if the timer is counting up or down. Valid Timer Reload values are 0001h - FFFFh. If the timer is running, the reload value will not be updated until the timer overflows or underflows. Programming a 0000h as a preload value is not a valid count value. Using a value of 0000h will cause unpredictable behavior.	R/W	FFFFh	Reset_ Timer

17.11.4 TIMER X COUNT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	TIMER_COUNT The Timer Count register returns the current value of the timer in all modes.	R	FFFFh	Reset_ Timer

© 2021 Microchip Technology Inc.

18.0 INPUT CAPTURE AND COMPARE TIMER

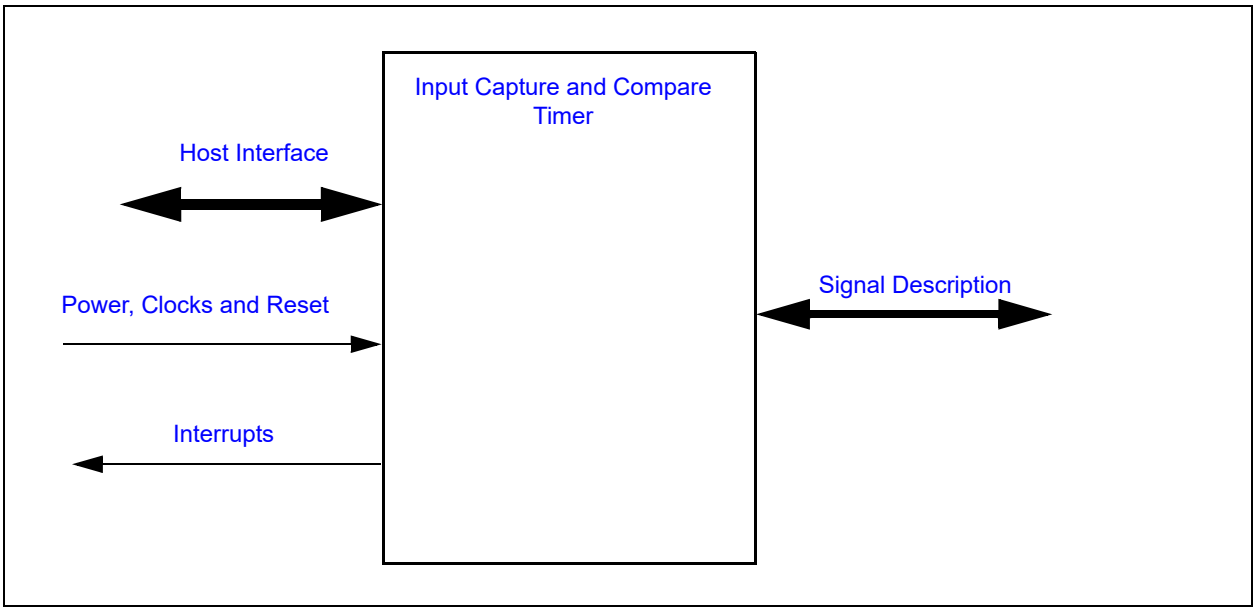
18.1 Introduction

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.

18.2 Interface

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

FIGURE 18-1: I/O DIAGRAM OF BLOCK



18.3 Signal Description

TABLE 18-1: 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.		

18.4 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".

18.5 Power, Clocks and Reset

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

18.5.1 POWER DOMAINS

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

18.5.2 CLOCK INPUTS

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

18.5.3 RESETS

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

18.6 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.

18.7 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.

© 2021 Microchip Technology Inc.

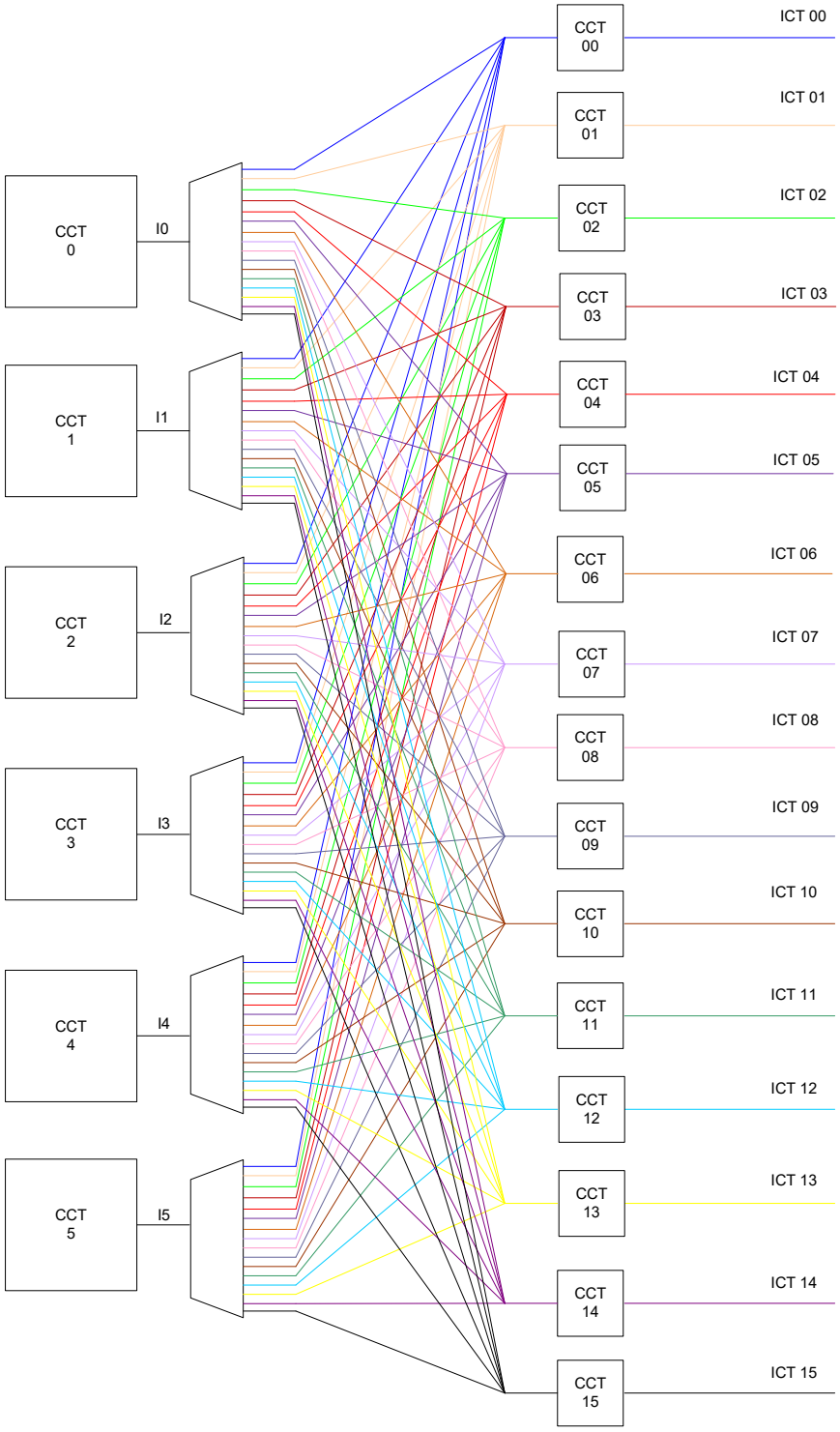
When the block returns from sleep, if enabled, the Free Running Timer Register value will continue counting from where it was when the block entered the Sleep state.

18.8 Description

The Input Capture and Compare Timer block has ICT Channel inputs and these can be connected to any of the 6 Capture Compare timer as shown in **FIGURE 18-2: "Capture and Compare Timer Port Connectivity"**. Please refer Table 1-1, "EEC1727 Feature List" for number of ICT channels present in the package.

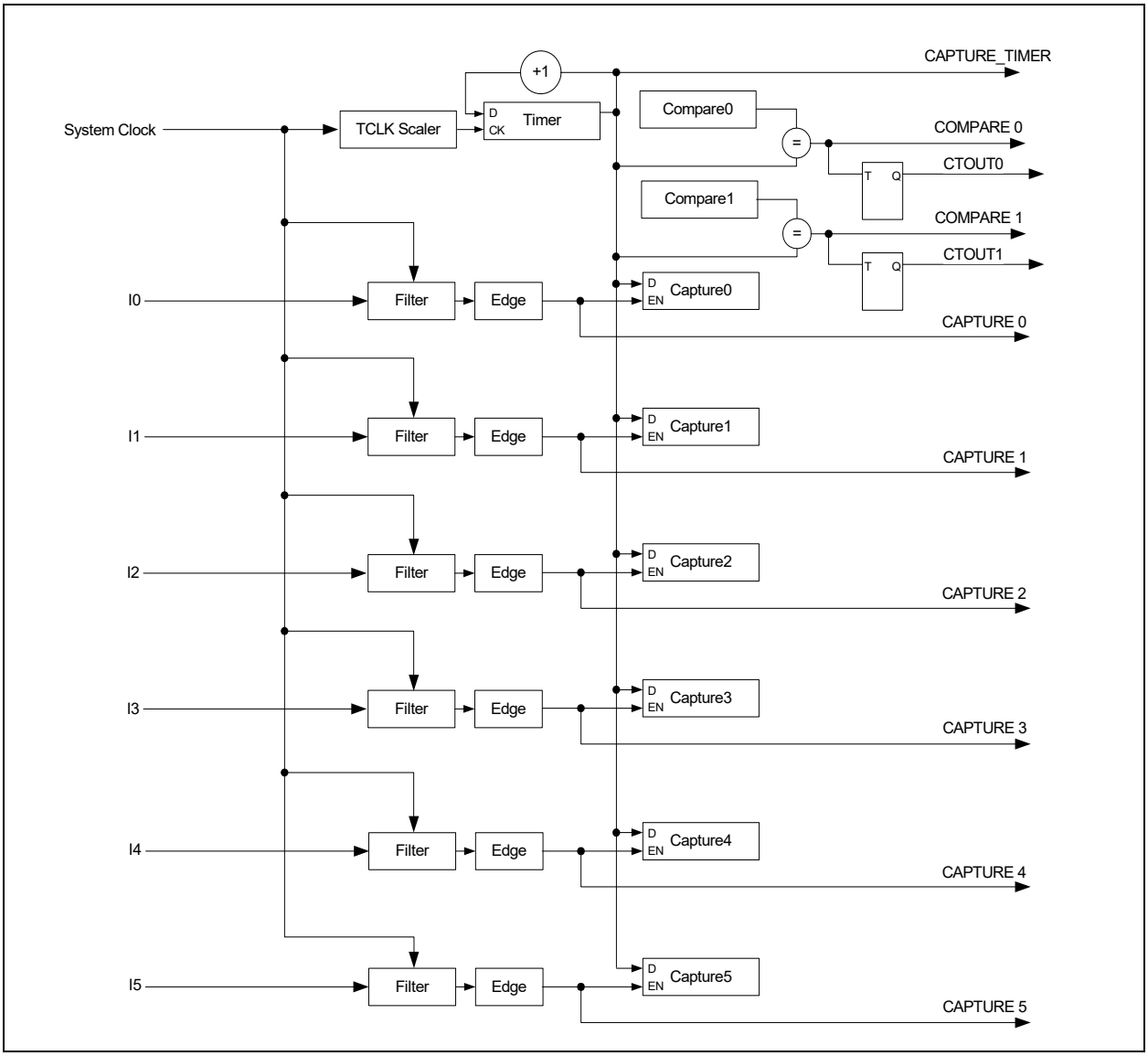
Note:	The CCT0 to CCT5 blocks shown in **FIGURE 18-2: "Capture and Compare Timer Port Connectivity"** are expanded and shown in **FIGURE 18-3: "Capture and Compare Timer Block Diagram"**

FIGURE 18-2: CAPTURE AND COMPARE TIMER PORT CONNECTIVITY



© 2021 Microchip Technology Inc.

FIGURE 18-3: CAPTURE AND COMPARE TIMER BLOCK DIAGRAM



18.8.1 TIMER CLOCK

Any of the frequencies listed in Table 18-2 may be used as the time base for the Free Running Counter.

TABLE 18-2: 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

TABLE 18-2: TIMER CLOCK FREQUENCIES (CONTINUED)

Timer Clock Select	Frequency Divide Select	Frequency Selected
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.

18.8.2 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.

18.9 Operation

18.9.1 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.

18.9.2 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.

18.9.2.1 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.

18.9.2.2 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.

18.10 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.

© 2021 Microchip Technology Inc.

TABLE 18-3: 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

18.10.1 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
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	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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 18-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
3	Reserved	RES	-	-
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

© 2021 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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

18.10.2 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 18-2, "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
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 18-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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 18-2, "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
7:5	FCLK_SEL0 This 3-bit field sets the clock source for the input filter for Capture Register 0. See Table 18-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
4:3	Reserved	RES	-	-

© 2021 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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

18.10.3 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 18-2, "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
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 18-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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

18.10.4 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

18.10.5 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

© 2021 Microchip Technology Inc.

18.10.6 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. Note 1	R	0h	RESET_SYS
Note 1: Any ICT input can be routed to any capture register using the ICT mux select register				

18.10.7 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. Note 1	R	0h	RESET_SYS

18.10.8 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. Note 1	R	0h	RESET_SYS

18.10.9 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. Note 1	R	0h	RESET_SYS

© 2021 Microchip Technology Inc.

18.10.10 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. Note 1	R	0h	RESET_SYS

18.10.11 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

18.10.12 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

18.10.13 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

© 2021 Microchip Technology Inc.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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

19.0 HIBERNATION TIMER

19.1 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.

19.2 References

No references have been cited for this chapter.

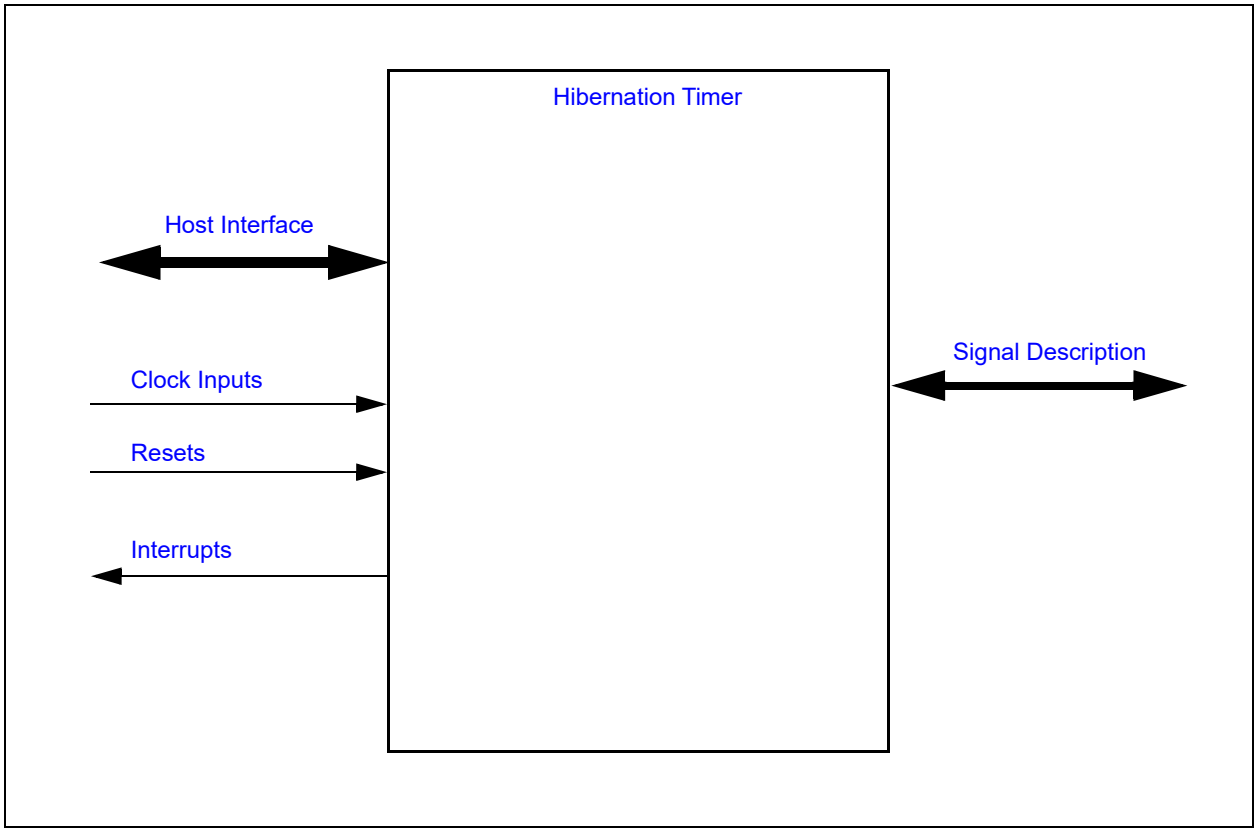
19.3 Terminology

No terms have been cited for this chapter.

19.4 Interface

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

FIGURE 19-1: HIBERNATION TIMER INTERFACE DIAGRAM



19.5 Signal Description

There are no external signals for this block.

19.6 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".

© 2021 Microchip Technology Inc.

19.7 Power, Clocks and Reset

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

19.7.1 POWER DOMAINS

TABLE 19-1: POWER SOURCES

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

19.7.2 CLOCK INPUTS

TABLE 19-2: CLOCK INPUTS

Name	Description
32KHz Core	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.

19.7.3 RESETS

TABLE 19-3: 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.

19.8 Interrupts

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

Each instance of the Hibernation Timer in the EEC1727 can be used to generate interrupts and wake-up events when the timer decrements to zero.

TABLE 19-4: 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.

19.9 Low Power Modes

The timer operates off of the 32KHz Core 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.

19.10 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 19-5: REGISTER SUMMARY

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

19.10.1 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

19.10.2 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

19.10.3 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

© 2021 Microchip Technology Inc.

20.0 RTOS TIMER

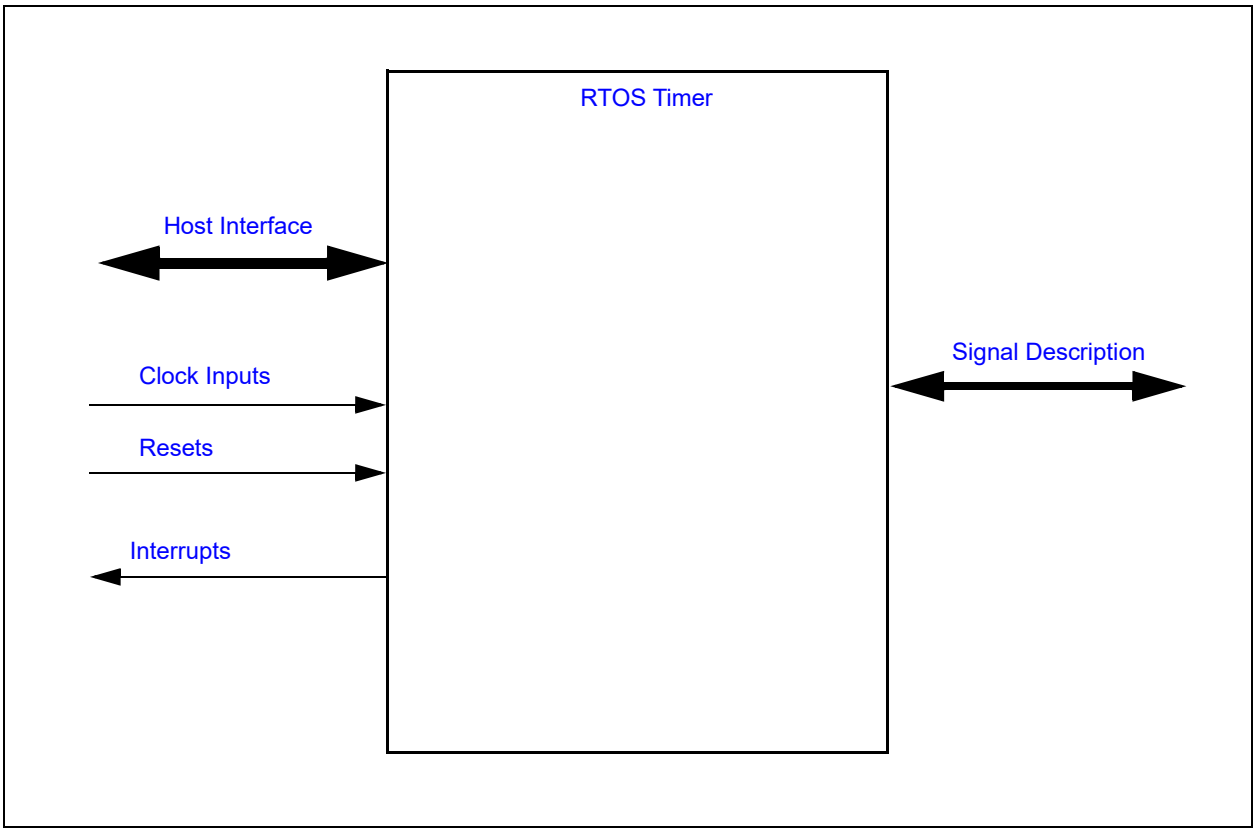
20.1 Introduction

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.

20.2 Interface

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

FIGURE 20-1: I/O DIAGRAM OF BLOCK



20.3 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.).

20.4 Host Interface

The Embedded Controller (EC) may access this block via the registers defined in Section 20.9, "EC Registers".

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 timer control logic and registers are all implemented on this single power domain.

20.5.2 CLOCK INPUTS

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

20.5.3 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.

20.6 Interrupts

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

20.7 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.

20.8 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.

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 RTOS Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

© 2021 Microchip Technology Inc.

TABLE 20-1: REGISTER SUMMARY

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

20.9.1 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

20.9.2 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

20.9.3 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

© 2021 Microchip Technology Inc.

20.9.4 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

© 2021 Microchip Technology Inc.

21.0 REAL TIME CLOCK

21.1 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.

21.2 References

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

21.3 Terminology

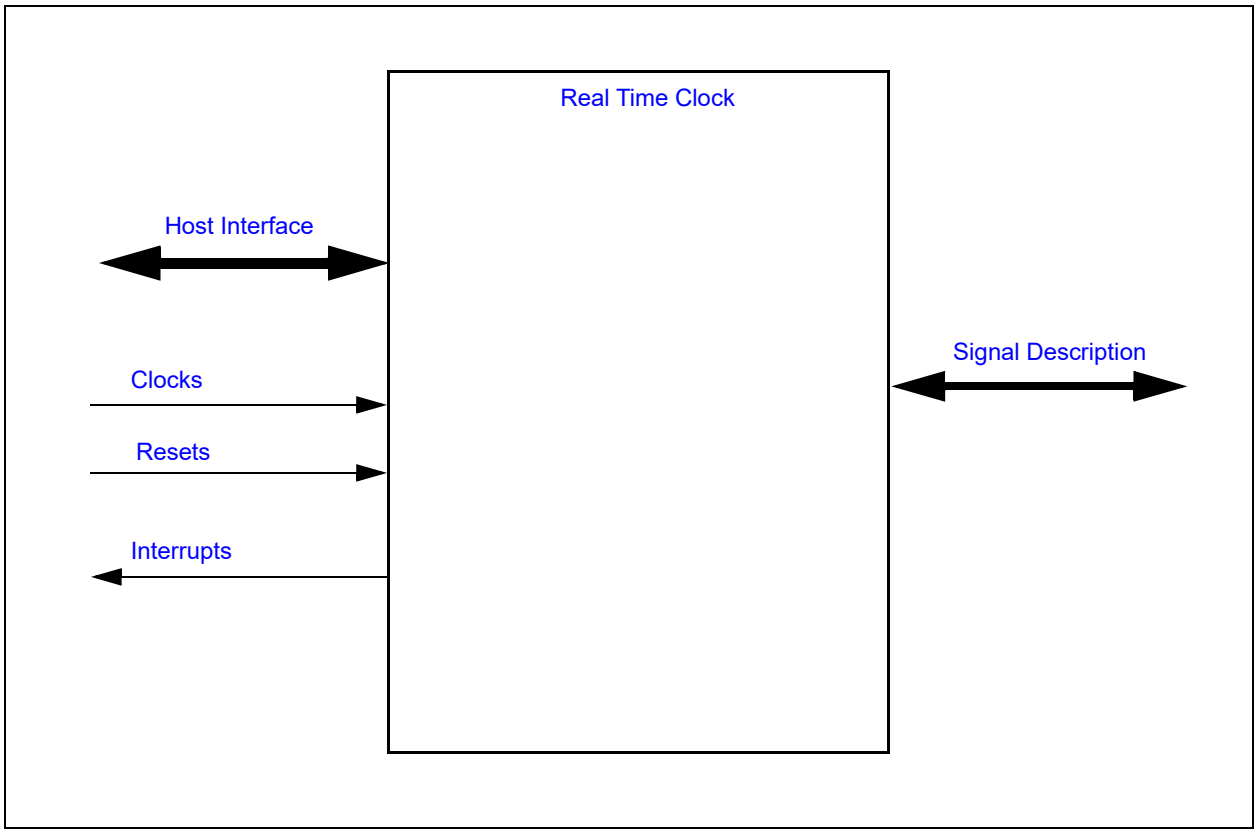
Time and Date Registers:

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**.

21.4 Interface

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

FIGURE 21-1: I/O DIAGRAM OF BLOCK



© 2021 Microchip Technology Inc.

21.5 Signal Description

There are no external signals.

21.6 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".

21.7 Power, Clocks and Resets

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

21.7.1 POWER DOMAINS

TABLE 21-1: 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.

21.7.2 CLOCKS

TABLE 21-2: CLOCKS

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

21.7.3 RESETS

TABLE 21-3: 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

21.8 Interrupts

TABLE 21-4: 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 21-5: 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.

21.9 Low Power Modes

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

21.10 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.

21.11 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 21-6: 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

© 2021 Microchip Technology Inc.

TABLE 21-6: 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.

21.11.1 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

21.11.2 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

21.11.3 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

21.11.4 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

21.11.5 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

21.11.6 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

© 2021 Microchip Technology Inc.

21.11.7 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

21.11.8 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

21.11.9 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

21.11.10 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

© 2021 Microchip Technology Inc.

21.11.11 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 21-7	R/W	0h	RESET_RTC

TABLE 21-7: 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

© 2021 Microchip Technology Inc.

21.11.12 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.

21.11.13 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	-	-

21.11.14 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

© 2021 Microchip Technology Inc.

21.11.15 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

21.11.16 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

21.11.17 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.

21.11.18 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	-	-

© 2021 Microchip Technology Inc.

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.

22.0 WEEK TIMER

22.1 Introduction

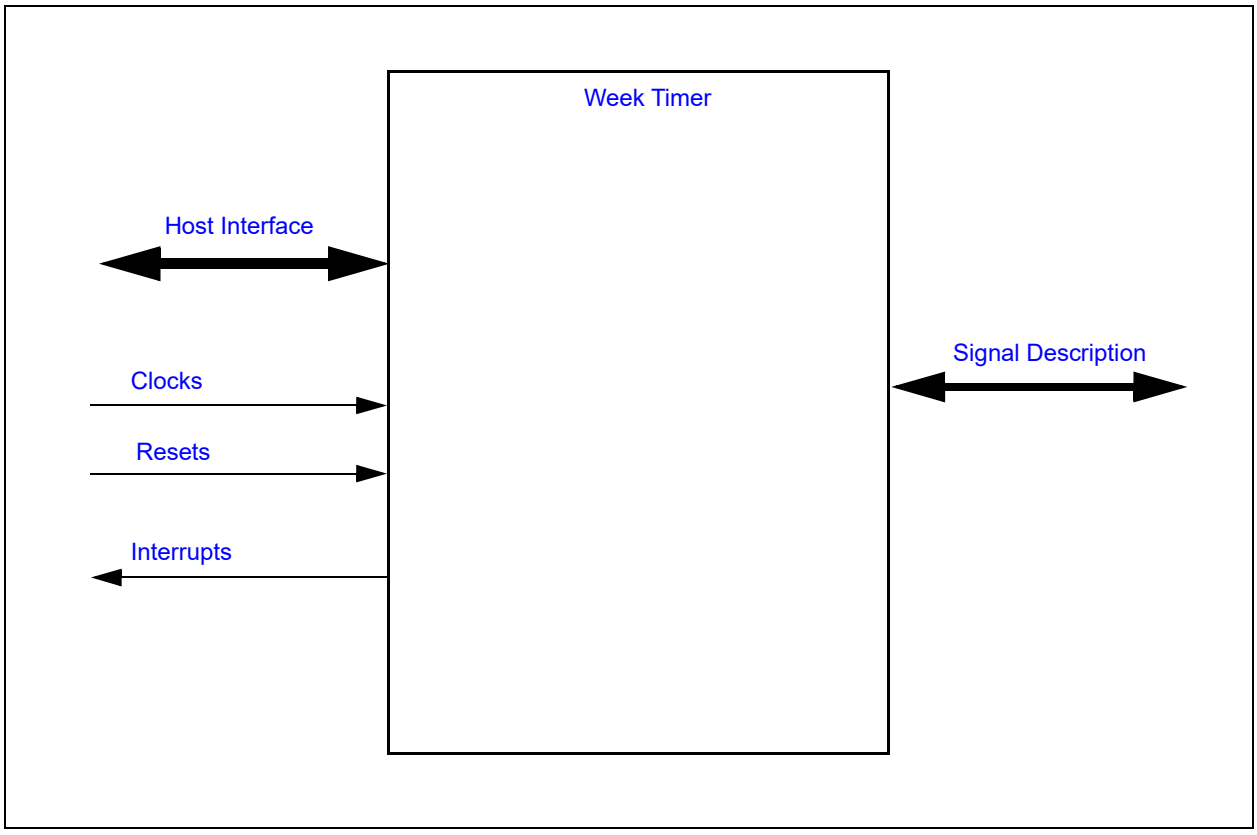
The Week Alarm Interface provides two timekeeping functions: a Week Timer and a Sub-Week Timer. Both the Week Timer and the Sub-Week Timer assert the Power-Up Event Output which automatically powers-up the system from the G3 state. Features include:

- 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

22.2 Interface

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

FIGURE 22-1: I/O DIAGRAM OF BLOCK



22.3 Signal Description

TABLE 22-2: INTERNAL SIGNAL DESCRIPTION TABLE

Name	Direction	Description
POWER_UP_EVENT	OUTPUT	Signal to the VBAT-Powered Control Interface. When this signal is asserted, the VCI output signal asserts. See Section 22.8, "Power-Up Events".

© 2021 Microchip Technology Inc.

22.4 Host Interface

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

22.5 Power, Clocks and Resets

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

22.5.1 POWER DOMAINS

TABLE 22-3: 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.

22.5.2 CLOCKS

TABLE 22-4: CLOCKS

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

22.5.3 RESETS

TABLE 22-5: 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.

22.6 Interrupts

TABLE 22-6: 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.

TABLE 22-6: EC INTERRUPTS (CONTINUED)

Source	Description
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 22-9, "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.

22.7 Low Power Modes

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

22.8 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 is routed to the VBAT-Powered Control Interface (VCI). The VCI_OUT pin that is part of the VCI is asserted if the POWER_UP_EVENT is asserted.

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.

22.9 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.

22.9.1 INTERNAL COUNTERS

The Week Timer includes 3 counters:

22.9.1.1 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.

22.9.1.2 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.

© 2021 Microchip Technology Inc.

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 22-7: 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

22.9.1.3 15-bit Clock Divider

This counter is 15 bits wide. The clock for this counter is 32KHz Core, 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 22-9, "SPISR Encoding" for a list of available frequencies.

22.9.2 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.

22.9.3 APPLICATION NOTE: REGISTER TIMING

Register writes in the Week Alarm complete within two cycles of the 32KHz Core 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 Core clock are synchronously asserted to the EC.

22.9.4 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)));
}
```


© 2021 Microchip Technology Inc.

22.9.5 APPLICATION NOTE: WEEK TIMER INITIALIZATION AND PROGRAMMING SEQUENCE

The week alarm timers may sometimes expire earlier than the configured time interval by an interval of up to one unit of the alarm time source. For instance the when you configure the timer for an interval of 5 seconds with a base time unit of 1 second it may expire at an interval between 4 and 5 seconds.

In case of repeating or auto-reload alarms the issue may be observed only on the first instance of the expiry after enable.

No such deviation would be observed from the second expiry of the auto reload timer.

The issue can be avoided by introducing a 25uS delay before enabling the timer after writing to all other config registers of the timer. Any application which is sensitive/critical to the timer expiry is recommended to add a delay as suggested above before enabling the timer.

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 Week Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 22-8: 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

22.10.1 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 22.8, "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

22.10.2 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

© 2021 Microchip Technology Inc.

22.10.3 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

22.10.4 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

22.10.5 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 22-9, "SPISR Encoding" shows the relation between the SPISR encoding and Sub-Second interrupt rate.	R/W	00h	RESET_VBAT

TABLE 22-9: 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
4	16	62.5 ms
5	32	31.25 ms
6	64	15.63 ms

© 2021 Microchip Technology Inc.

TABLE 22-9: SPISR ENCODING (CONTINUED)

SPISR Value	Sub-Second Interrupt Rate, Hz	Interrupt Period
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

22.10.6 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 22-7, "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	-	-

© 2021 Microchip Technology Inc.

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

22.10.7 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

23.0 TACH

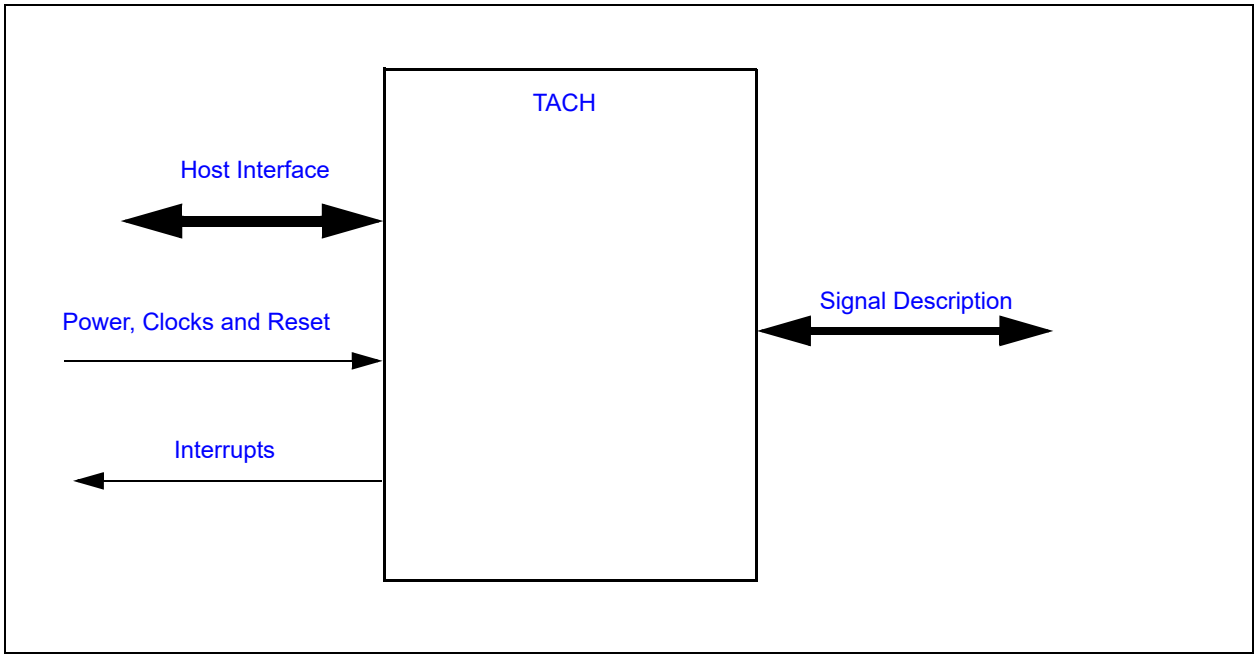
23.1 Introduction

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

23.2 Interface

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

FIGURE 23-1: I/O DIAGRAM OF BLOCK



23.3 Signal Description

TABLE 23-1: SIGNAL DESCRIPTION

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

23.4 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".

23.5 Power, Clocks and Reset

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

23.5.1 POWER DOMAINS

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

© 2021 Microchip Technology Inc.

23.5.2 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.

23.5.3 RESETS

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

23.6 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 23-2: 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.

23.7 Low Power Modes

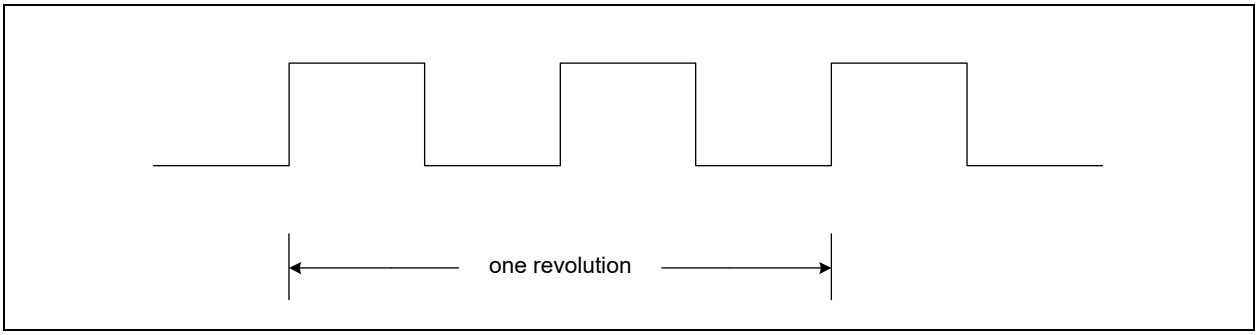
The TACH may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

23.8 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 23-2 below.

FIGURE 23-2: 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).

23.8.1 MODES OF OPERATION

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

23.8.1.1 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.

23.8.1.2 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.

23.8.2 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.

23.9 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 23-3: REGISTER SUMMARY

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

© 2021 Microchip Technology Inc.

23.9.1 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

© 2021 Microchip Technology Inc.

23.9.2 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.

23.9.3 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

23.9.4 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

© 2021 Microchip Technology Inc.

24.0 PWM

24.1 Introduction

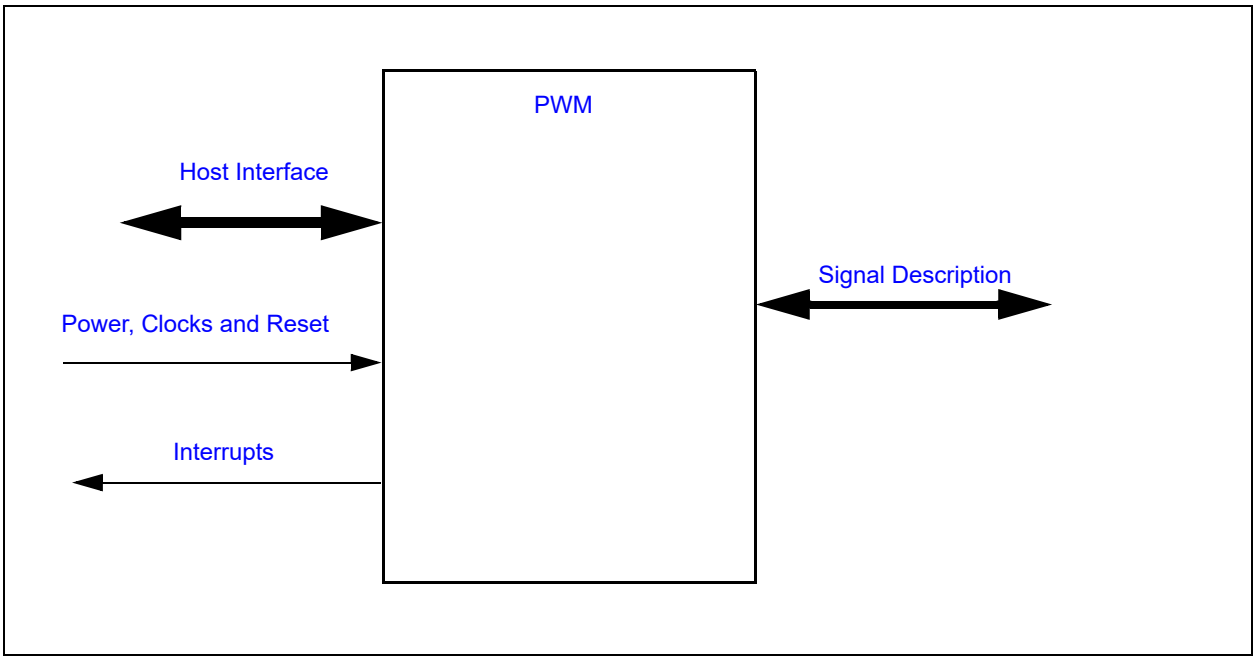
This block generates a PWM output that can be used to control 4-wire fans, blinking LEDs, and other similar devices. Each PWM can generate an arbitrary duty cycle output at frequencies from less than 0.1 Hz to 24 MHz.

The PWMx Counter ON Time registers and PWMx Counter OFF Time registers determine the operation of the PWM_OUTPUT signals. See Section 24.9.1, "PWMx Counter ON Time Register" and Section 24.9.2, "PWMx Counter OFF Time Register" for a description of the PWM_OUTPUT signals.

24.2 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



24.3 Signal Description

TABLE 24-1: SIGNAL DESCRIPTION

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

24.4 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".

24.5 Power, Clocks and Reset

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

24.5.1 POWER DOMAINS

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

24.5.2 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.

24.5.3 RESETS

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

24.6 Interrupts

The PWM block does not generate any interrupt events.

24.7 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.

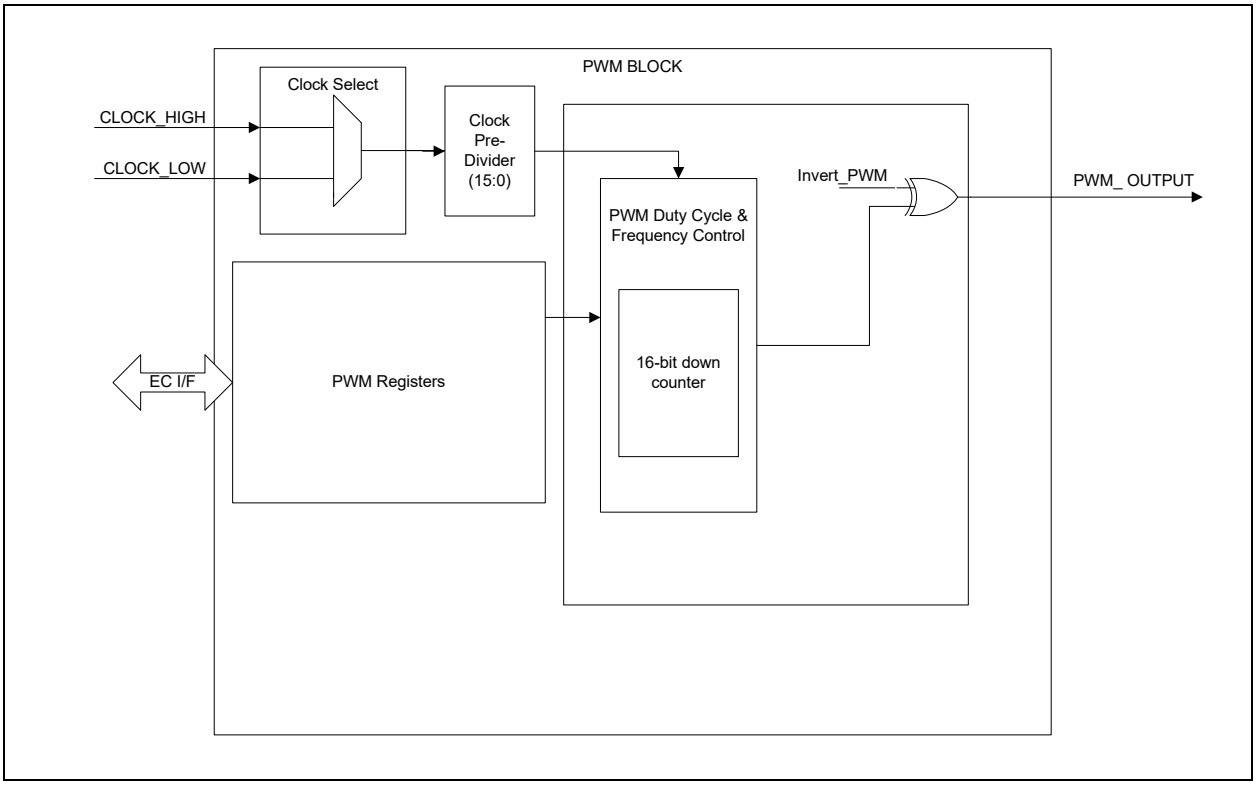
24.8 Description

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.

© 2021 Microchip Technology Inc.

FIGURE 24-2: BLOCK DIAGRAM OF PWM CONTROLLER



Note: In Figure 24-2, 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 24-1: 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 24.9, "EC Registers".

EQUATION 24-2: 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.

24.8.1 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.

24.9 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 24-2: REGISTER SUMMARY

Offset	Register Name
00h	PWMx Counter ON Time Register
04h	PWMx Counter OFF Time Register
08h	PWMx Configuration Register

24.9.1 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

© 2021 Microchip Technology Inc.

24.9.2 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

24.9.3 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

25.0 ANALOG TO DIGITAL CONVERTER

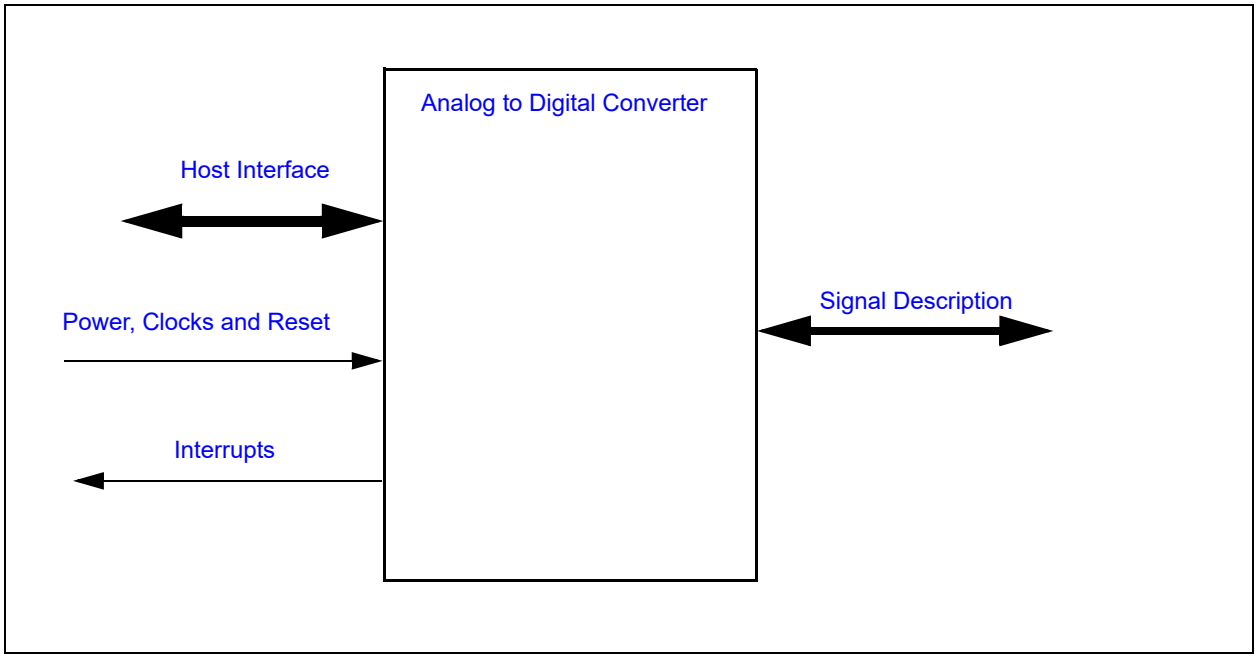
25.1 Introduction

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 with accuracy of +/- 4 LSB.

25.2 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



25.3 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
	Input	ADC Analog Voltage Input from pins. **Note:** The ADC Controller supports up to 16 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.

Note: GPIO pins adjacent to ADC input pins must not be toggled while ADC conversion is in progress.

© 2021 Microchip Technology Inc.

25.4 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".

25.5 Power, Clocks and Reset

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

25.5.1 POWER DOMAINS

TABLE 25-2: 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.

25.5.2 CLOCK INPUTS

TABLE 25-3: CLOCK INPUTS

Name	Description
48MHz	This clock signal is the master clock input to the ADC and may also be referred to as system clock in this chapter. 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.

25.5.3 RESETS

TABLE 25-4: 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.

25.6 Interrupts

TABLE 25-5: 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.

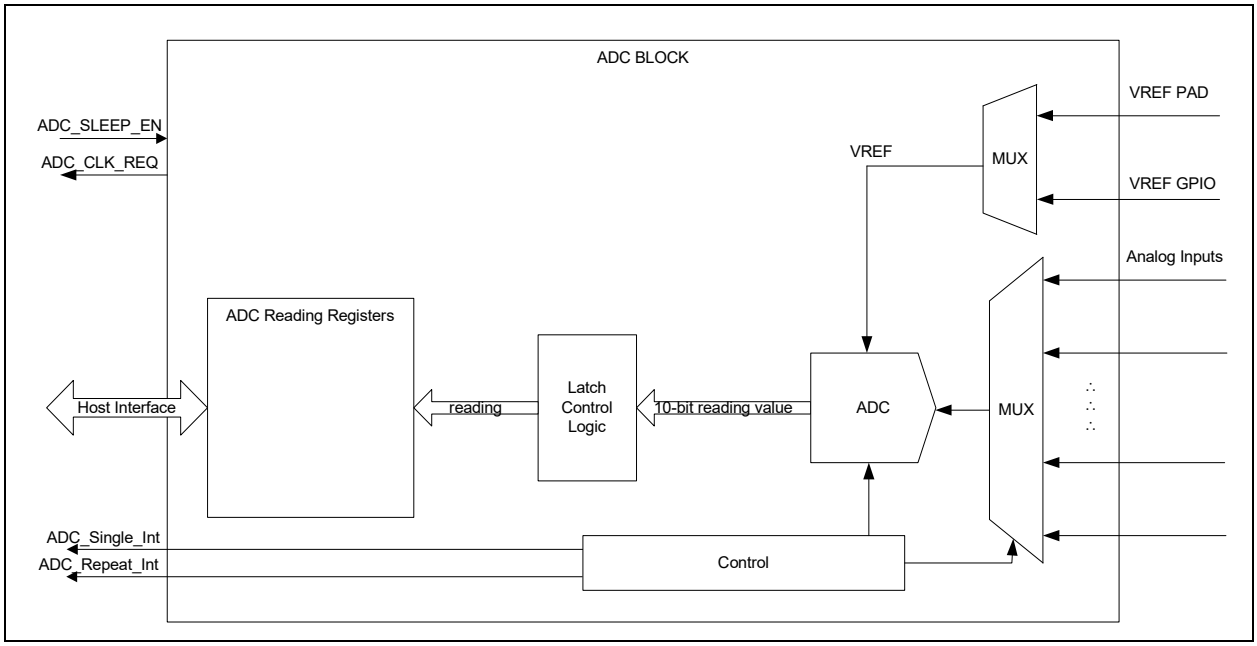
25.7 Low Power Modes

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.'

25.8 Description

FIGURE 25-2: ADC BLOCK DIAGRAM



The EEC1727 features a sixteen channel successive approximation Analog to Digital Converter. The ADC architecture features excellent linearity and converts analog signals to 12 bit words. A 12-bit conversion can be repeated as often as every 900ns. for any single channel, with the maximum ADC sampling clock setting of 24 MHz. The sixteen channels are implemented with a single high speed ADC fed by a sixteen input analog multiplexer. The multiplexer cycles through the sixteen voltage channels, starting with the lowest-numbered channel and proceeding to the highest-number channel at a fixed rate set by the Master Clock Input, dwelling only on those channels that are programmed to be active.

The input range on the voltage channels spans from 0V to the voltage reference. With a voltage reference of 3.3V, this provides resolutions of approximately 0.806 mV in 12-bit mode and 3.226 mV in 10-bit mode, respectively. 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 is 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.

Note:	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.

© 2021 Microchip Technology Inc.

25.8.1 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 START_DELAY in the ADC Delay Register and WARM_UP_DELAY in SAR ADC Control Register.Every channel that is enabled will be converted in 500ns for 12 bit mode and 416.6ns for 10bit mode, for 24MHz ADC sampling clock. The conversion time formula is **Resolution * Sampling clock time period.** This is the actual time between sampling of start of conversion (SOC) and assertion of end of conversion (EOC) excluding those two cycles. This does not include Warm Up delay, Startup delay, VREF switching delay and Charge delays which are user configurable.

- 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 conversion.
- After all channels enabled by RPT_EN are converted by the ADC, REPEAT_DONE_STATUS will be set to 1. The firmware must clear the REPEAT_DONE_STATUS bit for getting the interrupt for every repeat cycle.

Note:	**Total conversion time for one Repeat cycle** = START_DELAY + WARM_UP_DELAY + channel sequencing time of disabled channels (one 48MHz clock period per channel) + {(per channel conversion time + One ADC sampling clock + EOC settling time + Five 48MHz clocks period for Vref ready time) * (number of enabled channels)}.

Note:	The above **Total conversion time formula for one Repeat cycle** is showing the sequence of operations inside the ADC starting with START_DELAY and ending with number of enabled channels.
Note:	EOC settling time = (ADC_CLK_LOW_TIME + two 48MHz clocks period).
Note:	Vref ready time = Time required for the Vref (VREF_ADC) value to settle after each conversion.

25.8.2 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.

25.8.3 APPLICATION NOTES

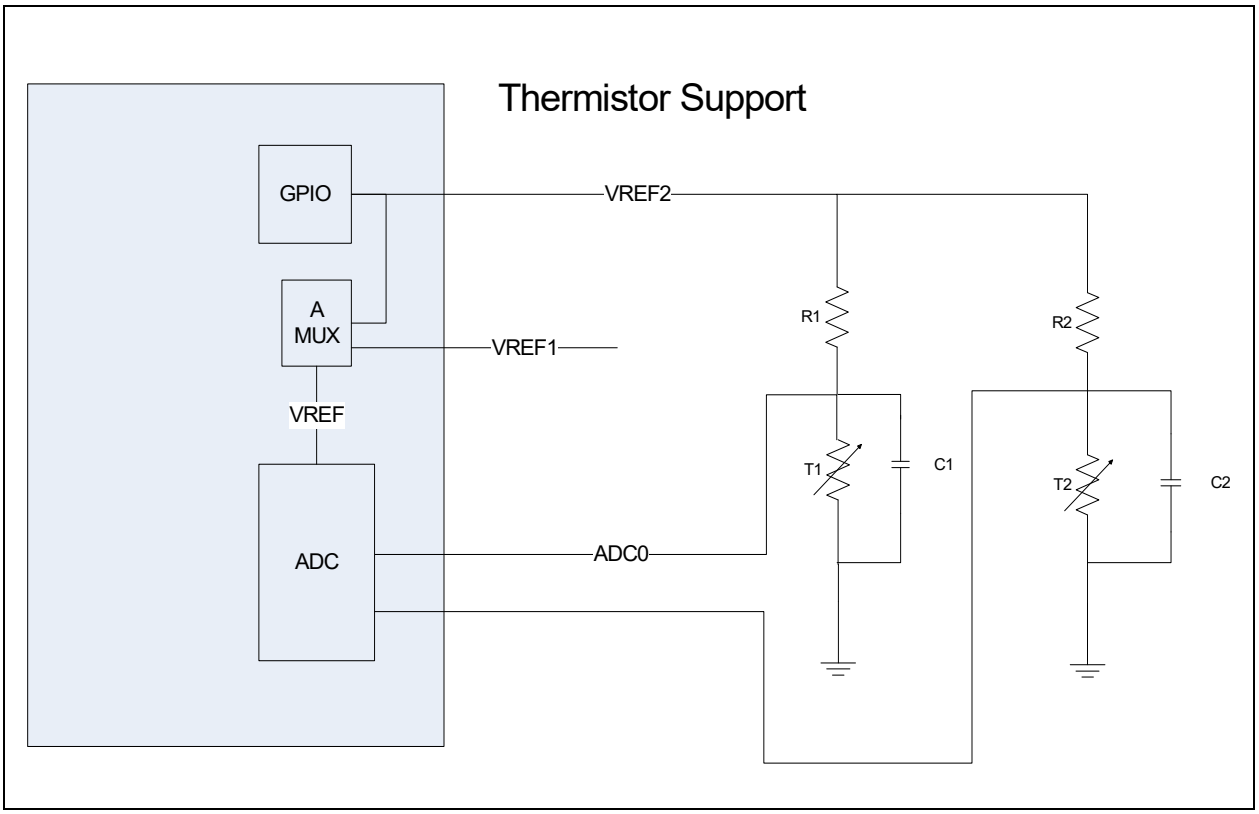
Please refer to white paper on “Accurate Temperature measurement using Thermistor” for details on how to use ADC for better than 1 degree C temperature measurement accuracy. Refer to FIGURE 25-3: ADC Reference Voltage Connection on page 318 for details of ADC reference voltage usage.

Note:	ADC inputs require at least a 0.1 uF capacitor to filter glitches.

Note 1:	It is recommended to use ADC sampling clock of 24MHz
2:	ADC sampling clock should not be configured to less than 3MHz
3:	Repeat delay is dependent on the input impedance and sampling rate and will have to be tuned accordingly
4:	ADC inputs require 0.1uF capacitors to filter glitches
5:	Resistors used in the ADC inputs should be 1% Tolerance resistors

© 2021 Microchip Technology Inc.

FIGURE 25-3: ADC REFERENCE VOLTAGE CONNECTION



25.9 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 25-6: 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
7Ch	ADC Configuration Register
80h	VREF Channel Register
84h	VREF Control Register

© 2021 Microchip Technology Inc.

TABLE 25-6: REGISTER SUMMARY

Offset	Register Name
88h	SAR ADC Control Register

25.9.1 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	-	-
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

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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

© 2021 Microchip Technology Inc.

25.9.2 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:** 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.	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

25.9.3 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

25.9.4 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

25.9.5 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

25.9.6 ADC CHANNEL READING REGISTERS

All ADC channels return their results into a 32-bit reading register. In each case the low 12 bits of the reading register return the result of the Analog to Digital conversion and the upper 22/20 bits return 0. Table 25-6, "Register Summary" shows the addresses of all the reading registers. For 10 bit ADC mode, SHIFT_DATA determines if the ADC reading is at bits [11:2] or [9:0]. For 12 bit ADC mode, SHIFT_DATA field has no impact on output and all lower 12 bits are valid.

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.

25.9.7 ADC CONFIGURATION REGISTER

Offset	7Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TEST	R	-	-

© 2021 Microchip Technology Inc.

Offset	7Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:8	ADC_CLK_HIGH_TIME High Time Count ADC Sampling 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 Sampling 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

25.9.8 VREF CHANNEL REGISTER

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:30	VREF Select for Channel 15 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
29:28	VREF Select for Channel 14 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
27:26	VREF Select for Channel 13 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
25:24	VREF Select for Channel 12 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
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

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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
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

© 2021 Microchip Technology Inc.

25.9.9 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. This counter runs on 48MHz clock.	R/W	0h	RESET_SYS
15:0	VREF Charge Delay This is the time delay required to charge the external VREF capacitor. This counter runs on 48MHz clock.	R/W	0h	RESET_SYS

25.9.10 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 Sampling clock cycles.	R/W	202h	RESET_SYS
6-4	Reserved	RES	-	-
3	SHIFT_DATA Right justify ADC output data for 10 bit ADC mode. This field has no effect in the 12 bit ADC mode. 0 = ADC_DOUT will be on bits [11:2] of ADC Channel Reading register for 10 bit ADC mode and lower bits [1:0] are 0 1 = ADC_DOUT will be on bits [9:0] of ADC Channel Reading register for 10 bit ADC mode as bits are 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

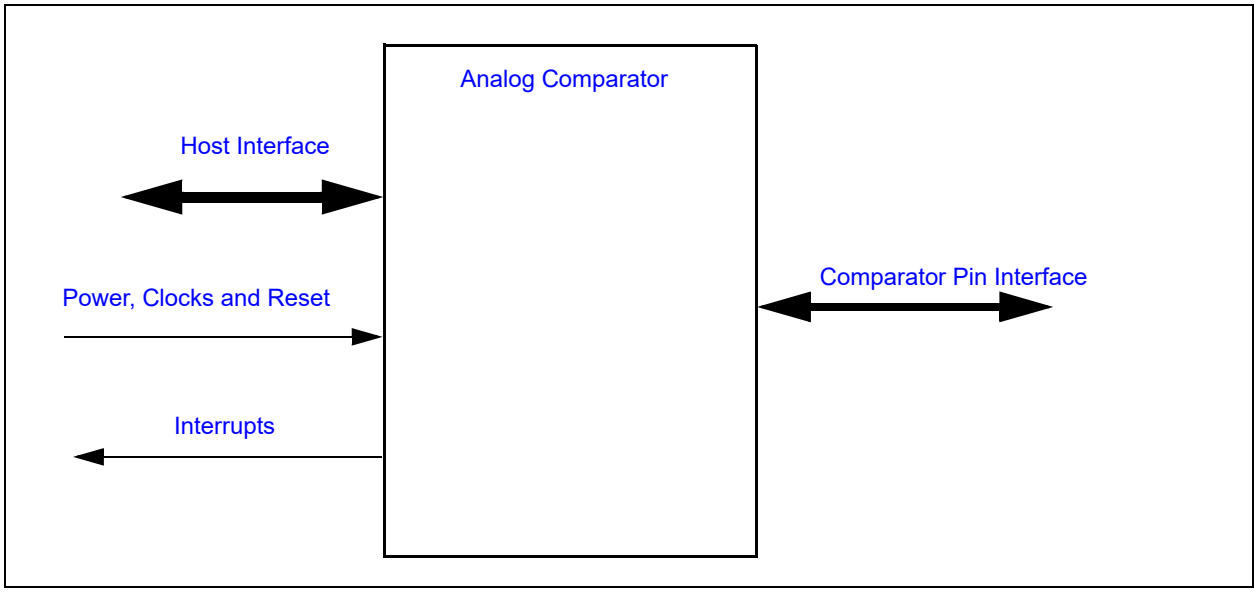
26.0 ANALOG COMPARATOR

26.1 Overview

26.2 The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison.**Interface**

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

FIGURE 26-1: I/O DIAGRAM OF BLOCK



26.3 Comparator Pin Interface

TABLE 26-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
CMP_VREF0	Input	Negative voltage input for Comparator 0
CMP_VREF1	Input	Negative voltage input for Comparator 1
CMP_VIN0	Input	Positive voltage input for Comparator 0
CMP_VIN1	Input	Positive voltage input for Comparator 1
CMP_VOUT0	Output	Comparator 0 output
CMP_VOUT1	Output	Comparator 1 output

26.4 Host Interface

The registers defined for the Comparator Interface are only accessible by the embedded controller. The Comparator Registers for both comparators are located in one register in the EC Subsystem register bank. See Section 33.8.15, "Comparator Control Register," on page 414.

© 2021 Microchip Technology Inc.

26.5 Power, Clocks and Reset

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

26.5.1 POWER DOMAINS

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

26.5.2 CLOCK INPUTS

This component does not require a clock input.

26.5.3 RESETS

Name	Description
RESET_VTR	This signal resets all the register in the EC Subsystem that interact with the comparators.

26.6 Interrupts

The comparators do not have a dedicated interrupt output event. An interrupt can be generated by the GPIO which shares the pin with the comparator output signal. Please refer to Section 2.3, "Pin List," on page 11 for the GPIO's that are mapped to the CMP_VOUTx functions.

The GPIO interrupt is configurable, thereby allowing CMP_VOUTx signal to generate an event when the CMP_VINx input is greater than the CMP_VREFx input or when it is less than the CMP_VREFx input. See the definition of Bits[7:4] of the Pin Control Registers on page 217.

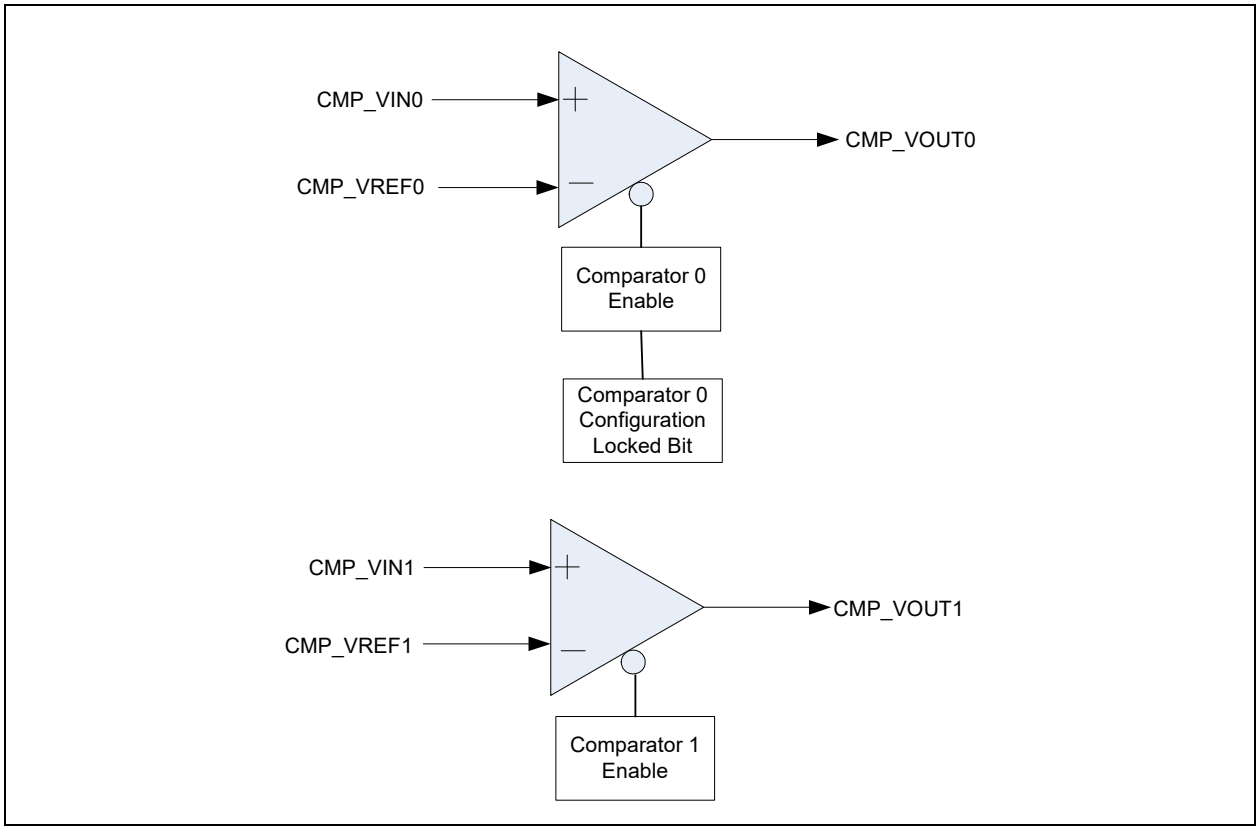
26.7 Low Power Modes

Each comparator is in its lowest powered state when its ENABLE bit is '0'.

26.8 Description

The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison. The reference voltage can be derived either from an external pin or from the internal Digital Analog Converter.

FIGURE 26-2: COMPARATOR BLOCK DIAGRAM



The Analog Comparator compares the analog voltage on the CMP_VINx input pin to a reference voltage and generates an output that indicates the result of the comparison. The reference voltage is derived from the CMP_VREFx input.

The GPIO that shares a pin with the CMP_VOUT signal can be used to generate an interrupt to the EC when the pin multiplexer is configured for CMP_VOUT. The GPIO Pin Control Register is configured for the desired interrupt behavior (level or edge). Changes in the CMP_VOUT output signal will be reflected in the Interrupt Status register field for the GPIO, as configured in the GPIO Pin Control Register.

The control bits for Comparator 0 can be locked. The COMPARATOR 0 ENABLE bit is locked if the LOCK bit for Comparator 0 is set. Once the LOCK bit is set, the COMPARATOR 0 ENABLE cannot be modified until the device is power cycled.

26.9 Comparator Registers

Control and status for both comparators are located in the EC Subsystem register bank. See Section 33.8.15, "Comparator Control Register," on page 414.

© 2021 Microchip Technology Inc.

27.0 RC IDENTIFICATION DETECTION (RC_ID)

27.1 Introduction

The Resistor/Capacitor Identification Detection (RC_ID) interface provides a single pin interface which can discriminate a number of quantized RC constants.

27.2 References

No references have been cited for this feature.

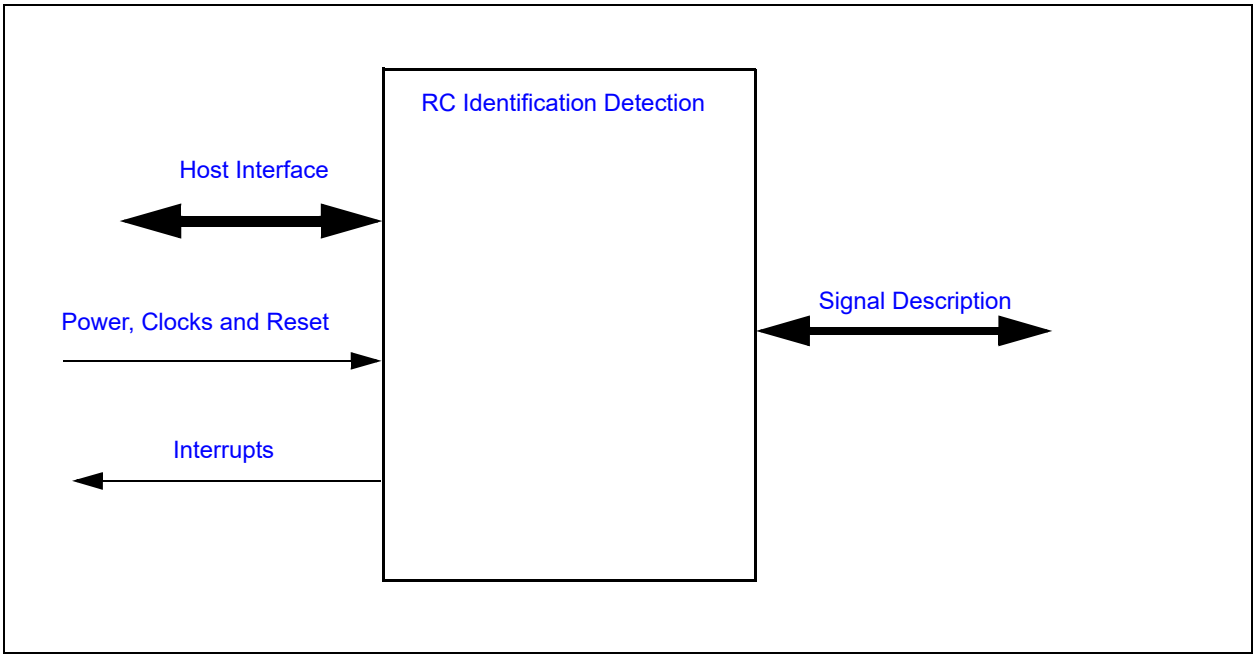
27.3 Terminology

There is no terminology defined for this section.

27.4 Interface

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

FIGURE 27-1: I/O DIAGRAM OF BLOCK



27.5 Signal Description

Name	Direction	Description
RC_ID	Input	Analog input used for measuring an external Resistor-Capacitor delay.

27.6 Host Interface

The registers defined for this block are accessible by the various hosts as indicated in Section 27.12, "EC Registers".

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 logic and registers implemented in this block are powered by this power well.

27.7.2 CLOCK INPUTS

Name	Description
48MHz	The main clock domain, used to generate the time base that measures the RC delay.

27.7.3 RESETS

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

27.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
RCID	This internal signal is generated when the DONE bit in the RC_ID Control Register is set to ‘1’.

27.9 Low Power Modes

This block may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry. If a measurement has been started, the block will continue to assert its clock_req output until the measurement completes.

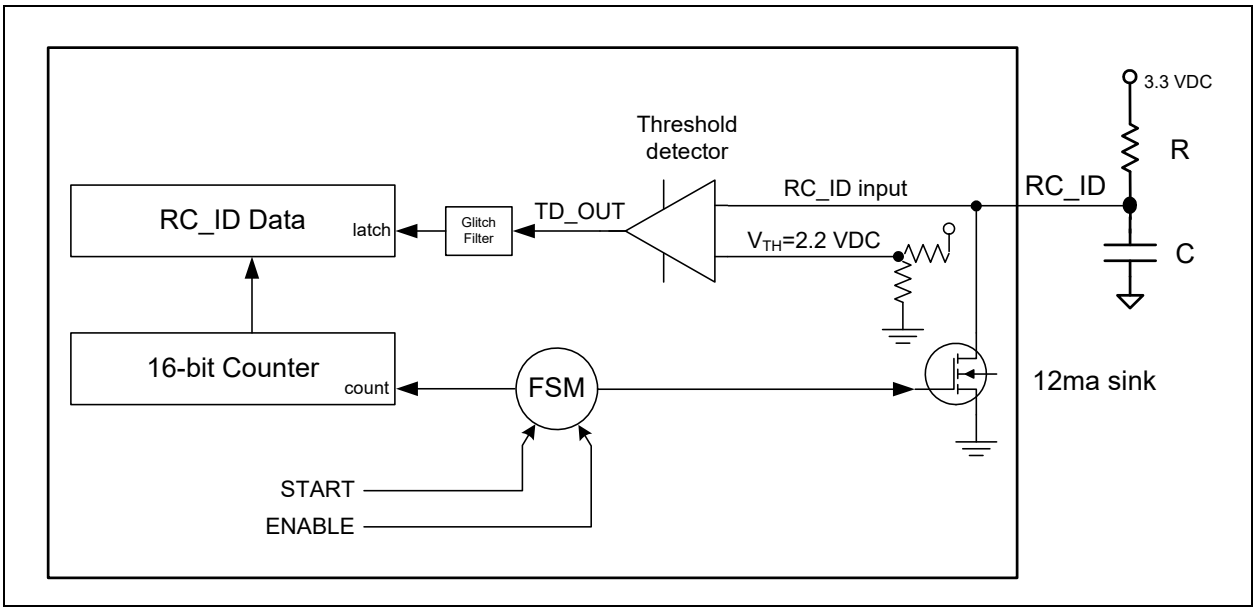
27.10 Description

Note: The RC_ID block only operates on 3.3V. The VTR pin associated with RC_ID signals must be connected to a 3.3V supply. If the VTR pin is supplied with 1.8V, the RC_ID logic will not function correctly.

The Resistor/Capacitor Identification Detection (RC_ID) interface provides a single pin interface which can discriminate a number of quantized RC constants. The judicious selection of RC values can provide a low cost means for system element configuration identification. The RC_ID I/O pin measures the charge/discharge time for an RC circuit connected to the pin as shown in Figure 27-2.

© 2021 Microchip Technology Inc.

FIGURE 27-2: BLOCK DIAGRAM OF RC Identification Detection (RC_ID)



The RC_ID interface determines the selected RC delay by measuring the rise time on the RC_ID pin that is attached to the RC circuit, as shown in the above figure. The measurement is performed by first discharging the external capacitor for a fixed period of time, set by an internal 16-bit counter running at a configurable time base, and then letting the capacitor charge again, using the same counter and time base to count how many clock ticks are required until the voltage on the capacitor exceeds 2.2V. A glitch filter, consisting of three ticks of the 48MHz main oscillator, smooths the threshold detection.

By fixing the capacitor value and varying the resistor value, up to eight discrete values can be determined based on the final count. Section 27.11, "Time Constants" shows a range of possible R and C values that can be used to create eight ID values.

Measurement requires five phases:

1. Reset. The two control bits (ENABLE and START) and the three status bits (TC, DONE and CY_ER) in the RC_ID Control Register are all '0'. The RD_IC pin is tri-stated and the block is in its lowest power state. In order to enter the Reset state, firmware must write the ENABLE, START and CLOCK_SET fields to '0' simultaneously or unpredictable results may occur.
2. Armed. Firmware enables the transition to this state by setting the ENABLE bit to '1' and the CLOCK_SET field to the desired time base. The START must remain at '0'. All three fields must be set with one write to the RC_ID Control Register. In this state the RC_ID clock is enabled and the 16-bit counter is armed. Firmware must wait a minimum of 300µS in the Armed phase before starting the Discharged phase.
3. Discharged. Firmware initiates the transition to the Discharged state by setting the ENABLE bit to '1', the START bit to '1' and the CLOCK_SET field to the desired clock rate, in a single write to the RC_ID Control Register. The RC_ID pin is discharged while the 16-bit counter counts from 0000h to FFFFh at the configured time base. When the counter reaches FFFFh the TC status bit is set to '1'. If at the end of the Discharged state the RC_ID pin remains above the 2.2V threshold, the CY_ER bit is set to '1', since the measurement will not be valid.
4. Charged.The RC_ID state machine automatically transitions to this state after the 16-bit counter reaches FFFFh while in the Discharged state. The 16-bit counter starts counting up from 0000h. The counter stops counting and its value is copied into the RC_ID Data Register when the voltage on the pin exceeds 2.2V. If the counter reaches FFFFh and the pin voltage remains below 2.2V, the CY_ER bit is set to '1'.
5. Done. After the counter stops counting, either because the pin voltage exceed the 2.2V threshold or the 16-bit counter reached FFFFh, the state machine transitions to this state. The DONE bit is set to '1' and the RC_ID interface re-enters its lowest power state. The interface will remain in the Done state until firmware explicitly initiates the Reset state.

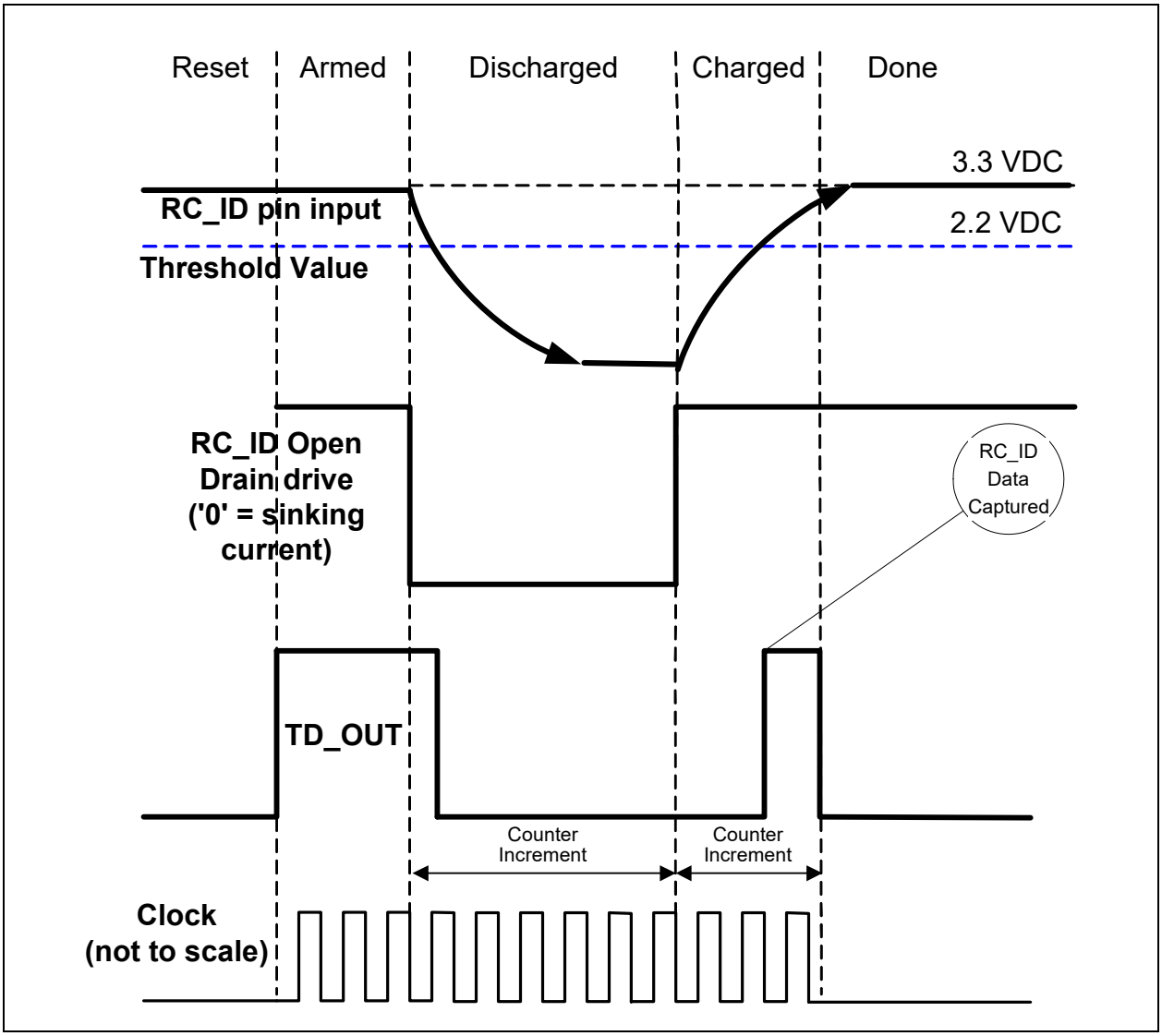
A new measurement must be started by putting the RC_ID Interface into the "Reset" state.

The five phases, along with the values of the control and status bits in the Control Register at the end of each phase, are summarized in the following table and figure:

TABLE 27-1: RC ID STATE TRANSITIONS

	State	ENABLE	START	TC	DONE
1.	Reset	0	0	0	0
2.	Armed	1	0	0	0
3.	Discharged	1	1	0	0
4.	Charged	1	1	1	0
5.	Done	1	1	1	1

FIGURE 27-3: RCID STATE TRANSITIONS



© 2021 Microchip Technology Inc.

27.11 Time Constants

This section lists a set of R and C values which can be connected to the RC_ID pin. Note that risetime generally follow RC time Tau. Firmware should use the Max and Min Counts in the tables to create quantized states.

In the following tables, the CLOCK_SET field in the RC_ID Control Register is set to '0', so the time base for measuring the rise time is 48MHz, the speed of the system clock. All capacitor values are ±10% and all resistor values are ±5%. Minimum and maximum count values are suggested ranges, calculated to provide reasonable margins around the nominal rise times. Rise times have been confirmed by laboratory measurements.

TABLE 27-2: SAMPLE RC VALUES, C=2200PF

R (KΩ)	Nominal Tau (µS)	Minimum Count	Maximum Count
1	2.2	60.00	72.00
2	4.4	115.00	140.00
4.3	9.5	241.00	294.00
8.2	18.04	456.00	557.00
33	72.6	1819.00	2224.00
62	136.4	3456.00	4224.00
130	286	7470.00	9130.00
240	528	14400.00	17600.00

TABLE 27-3: SAMPLE RC VALUES, C=3000PF

R (KΩ)	Nominal Tau (µS)	Minimum Count	Maximum Count
1	3	77.00	95.00
2	6	151.00	184.00
4.3	12.9	320.00	391.00
8.2	24.6	604.00	739.00
33	99	2439.00	2981.00
62	186	4647.00	5680.00
130	390	9990.00	12210.00
240	720	193508.00	23650.00

TABLE 27-4: SAMPLE RC VALUES, C=4700PF

R (KΩ)	Nominal Tau (µS)	Minimum Count	Maximum Count
1	4.7	116.00	142.00
2	9.4	229.00	280.00
4.3	20.2	495.00	605.00
8.2	38.5	945.00	1160.00
33	155.1	3780.00	4650.00
62	291.4	7249.00	8859.00
130	611	15480.00	18920.00
240	1128	29880.00	36520.00

© 2021 Microchip Technology Inc.

27.12 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 RC Identification Detection (RC_ID) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 27-5: REGISTER SUMMARY

Offset	Register Name
00h	RC_ID Control Register
04h	RC_ID Data Register

27.12.1 RC_ID CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	R	-	-
9:8	CLOCK_SET This field selects the frequency of the Counter circuit clock. This field must retain the same value as long as the ENABLE bit in this register is '1'. 3=6MHz 2=12MHz 1=24MHz 0=48MHz	R/W	0h	RESET_SYS
7	ENABLE Clearing the bit to '0' causes the RC_ID interface to enter the Reset state, gating its clocks, clearing the status bits in this register and entering into its lowest power state. Setting this bit to '1' causes the RC_ID interface to enter the Armed phase of an RC_ID measurement. When this bit is cleared to '0', the CLOCK_SET and START fields in this register must also be cleared to '0' in the same register write.	R/W	0h	RESET_SYS
6	START Setting this bit to '1' initiates the Discharged phase of an RC_ID measurement. Writes that change this bit from '0' to '1' must also write the ENABLE bit to '1', and must not change the CLOCK_SET field. A period of at least 300µS must elapse between setting the ENABLE bit to '1' and setting this bit to '1'.	R/W	0h	RESET_SYS
5:3	Reserved	R	-	-

© 2021 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	CY_ER This bit is ‘1’ if an RC_ID measurement encountered an error and the reading in the RC_ID Data Register is invalid. This bit is cleared to ‘0’ when the RC_ID interface is in the Reset phase. It is set either if during the Discharged phase the RC_ID pin did not fall below the 2.2V threshold, or if in the Charged phase the RC_ID pin did not rise above the 2.2V threshold and the 16-bit counter ended its count at FFFFh.	R	0h	RESET_SYS
1	TC This bit is cleared to ‘0’ when the RC_ID interface is in the Reset phase, and set to ‘1’ when the interface completes the Discharged phase of an RC_ID measurement.	R	0h	RESET_SYS
0	DONE This bit is cleared to ‘0’ when the RC_ID interface is in the Reset phase, and set to ‘1’ when the interface completes an RC_ID measurement.	R	0h	RESET_SYS

27.12.2 RC_ID DATA REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	DATA Reads of this register provide the result of an RC_ID measurement.	R	0h	RESET_SYS

28.0 BLINKING/BREATHING LED

28.1 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:

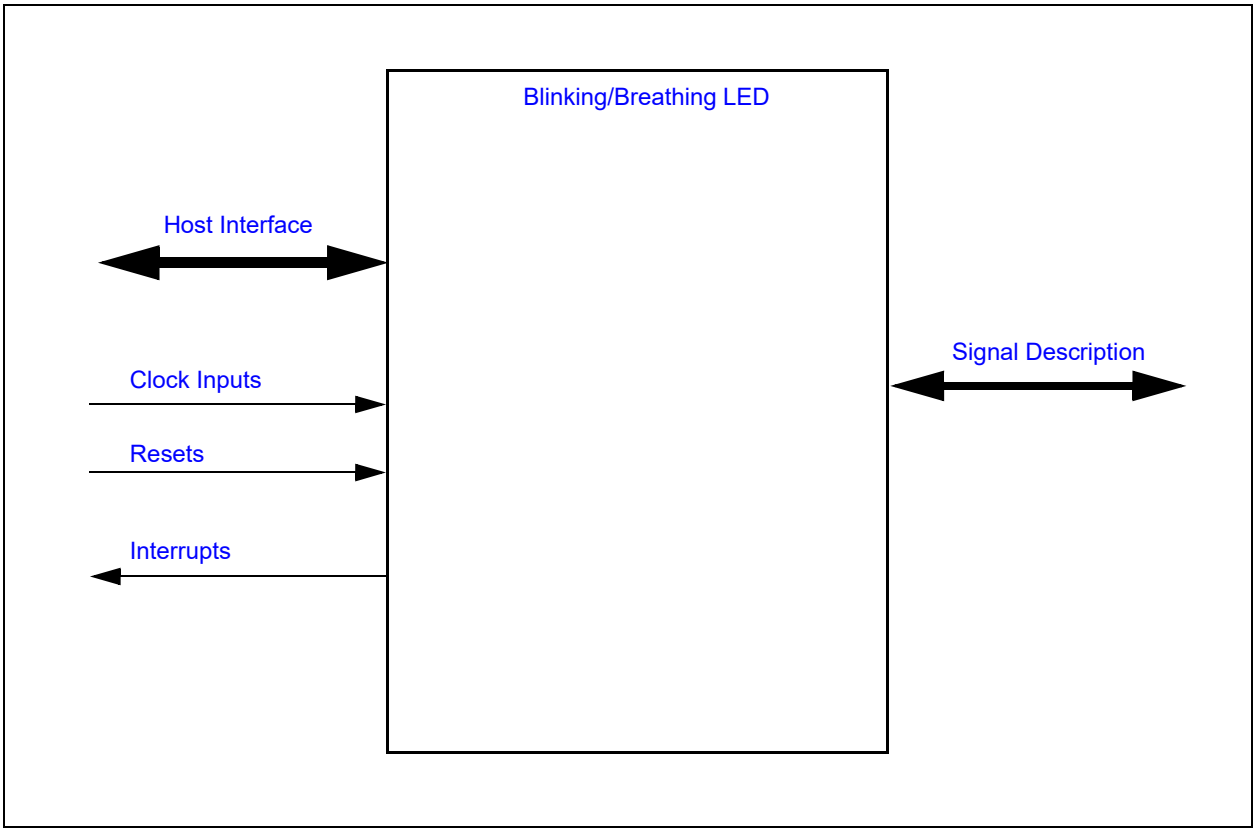
- Each PWM independently configurable
- Each PWM configurable for LED blinking and breathing output
- Highly configurable breathing rate from 60ms to 1min
- Non-linear brightness curves approximated with 8 piece wise-linear segments
- All LED PWMs can be synchronized
- Each PWM configurable for 8-bit PWM support
- Multiple clock rates
- Configurable Watchdog Timer

28.2 Interface

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

© 2021 Microchip Technology Inc.

FIGURE 28-1: I/O DIAGRAM OF BLOCK



28.3 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, "EEC1727 Feature List" table to know the number of LED pins available in the chip.

28.4 Host Interface

The blinking/breathing PWM block is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

28.5 Power, Clocks and Reset

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

28.5.1 POWER DOMAINS

Name	Description
VTR_CORE	Main power. The source of main power for the device is system dependent.

28.5.2 CLOCK INPUTS

Name	Description
32KHz Core	32.768 KHz clock
48MHz	Main system clock

28.5.3 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.

28.6 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 28.8.3.1, "PWM WDT".

Source	Description
PWM_WDT	PWM watchdog time out

28.7 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 28-1: 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 EEC1727 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.

© 2021 Microchip Technology Inc.

28.8 Description

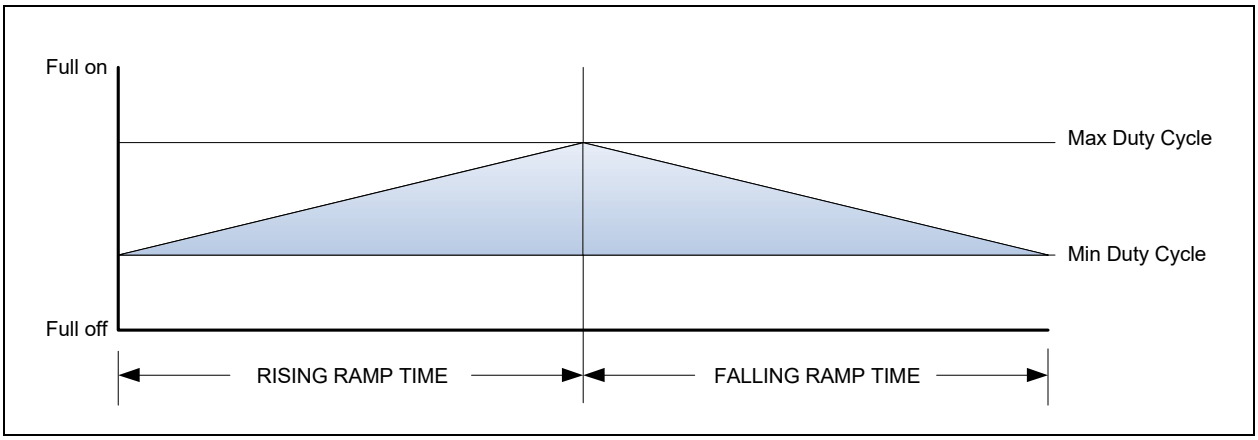
28.8.1 BREATHING

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:

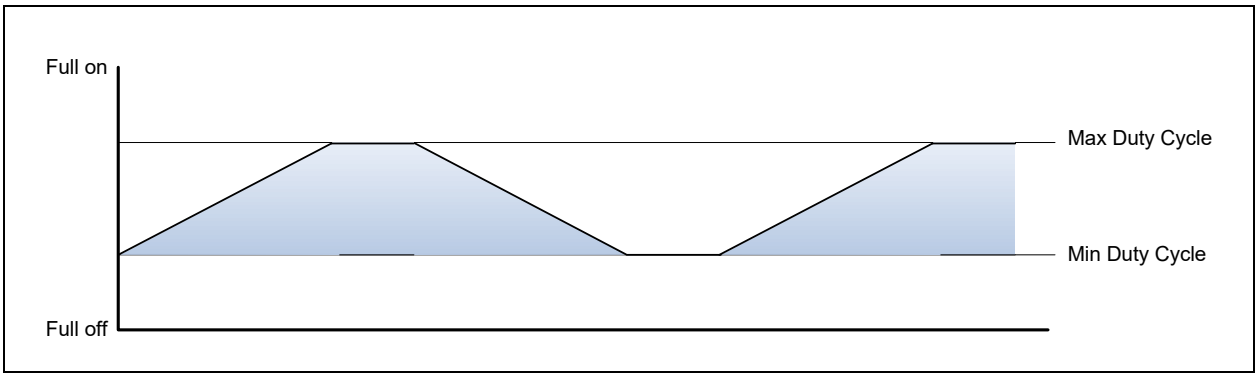
FIGURE 28-2: BREATHING LED EXAMPLE



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:

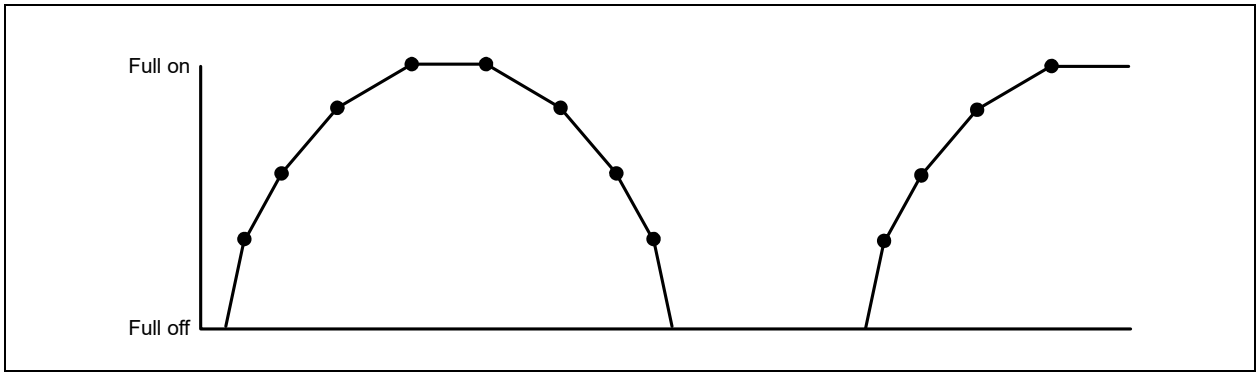
FIGURE 28-3: CLIPPING EXAMPLE



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 28-4: 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.

28.8.2 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 28-2, "LED Blink Configuration Examples" shows some example blinking configurations:

TABLE 28-2: 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.

© 2021 Microchip Technology Inc.

TABLE 28-3: BLINKING MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(32KHz Core frequency) /(PRESCALE + 1)/256
'H' Width	Seconds	(1/Frequency) x (DutyCycle/256)
'L' Width	Seconds	(1/Frequency) x ((1-DutyCycle)/256)

28.8.3 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 28-4.

TABLE 28-4: 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 28-5: 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)

28.8.3.1 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

28.9 Implementation

In addition to the registers described in Section 28.10, "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.

28.9.1 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 28-2, "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 28-6): 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 28-6).

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 28.10, "EC Registers", as well as the examples in Section 28.9.3, "Breathing Examples" for information on how to set these fields.

TABLE 28-6: SYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 28-3, "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 28-7: ASYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 28-3, "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]

© 2021 Microchip Technology Inc.

TABLE 28-7: ASYMMETRIC BREATHING MODE REGISTER USAGE (CONTINUED)

Rising/ Falling Ramp Times in Figure 28-3, "Clipping Example"	Duty Cycle	Segment Index	Asymmetric Mode Register Fields Utilized	
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 28-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].				

28.9.2 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.

28.9.3 BREATHING EXAMPLES

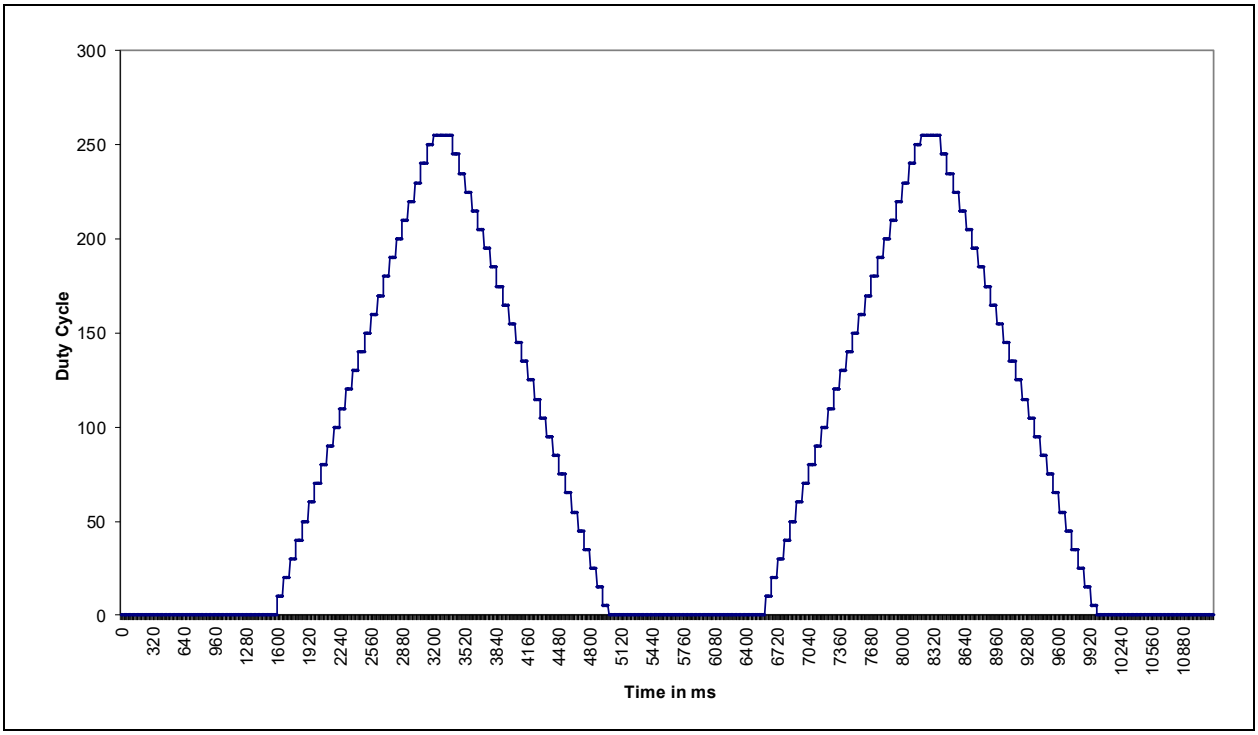
28.9.3.1 Linear LED brightness change

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 28-8: LINEAR EXAMPLE CONFIGURATION

Field	Value							
PSIZE	8-bit							
MAX	255							
MIN	0							
HD	25 ticks (200ms)							
LD	200 ticks (1.6s)							
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 28-5: LINEAR BRIGHTNESS CURVE EXAMPLE



28.9.3.2 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:

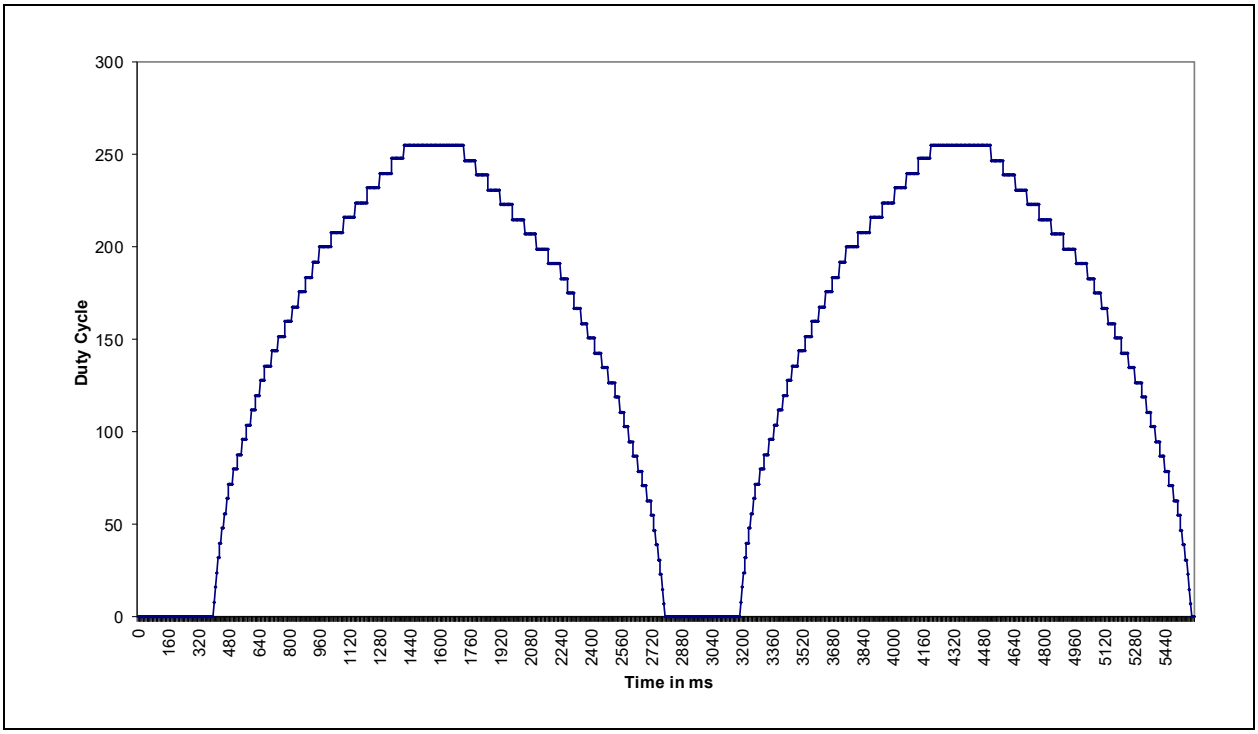
TABLE 28-9: NON-LINEAR EXAMPLE CONFIGURATION

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

© 2021 Microchip Technology Inc.

The resulting curve is shown in the following figure:

FIGURE 28-6: NON-LINEAR BRIGHTNESS CURVE EXAMPLE



28.10 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 28-10: 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 breath-

ing 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.

28.10.1 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 28-7, "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 28-2, "Breathing LED Example") are in Symmetric mode. Table 28-6, "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

© 2021 Microchip Technology Inc.

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

28.10.2 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

28.10.3 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

28.10.4 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 28-3, "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

...

© 2021 Microchip Technology Inc.

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

28.10.5 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 28-3, "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

© 2021 Microchip Technology Inc.

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

28.10.6 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

29.0 RPM-PWM INTERFACE

29.1 Introduction

The RPM-PWM Interface is a closed-loop RPM based Fan Control Algorithm that monitors a fan's speed and automatically adjusts the drive to the fan in order to maintain the desired fan speed.

The RPM-PWM Interface functionality consists of a closed-loop "set-and-forget" RPM-based fan controller.

29.2 References

No references have been cited for this chapter

29.3 Terminology

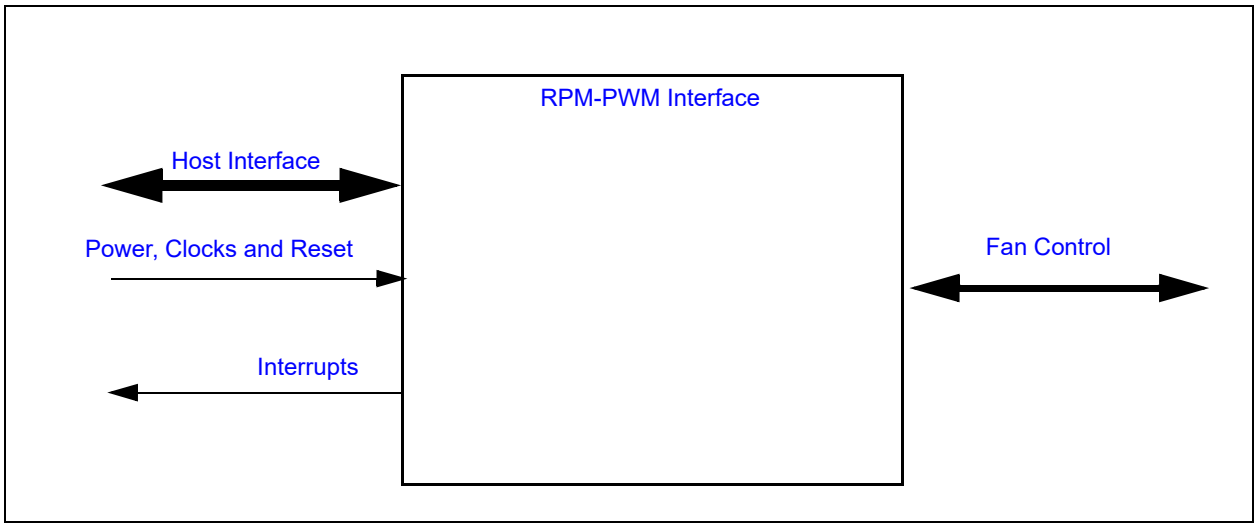
There is no terminology defined for this chapter.

29.4 Interface

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

The registers in the block are accessed by embedded controller code at the addresses shown in Section 29.9, "EC Registers".

FIGURE 29-1: RPM-PWM INTERFACE I/O DIAGRAM



29.4.1 FAN CONTROL

The Fan Control Signal Description Table lists the signals that are routed to/from the block.

Name	Direction	Description
GTACH	Input	Tachometer input from fan
GPWM	Output	PWM fan drive output

29.4.2 HOST INTERFACE

The registers defined for the RPM-PWM Interface are accessible by the various hosts as indicated in Section 29.9, "EC Registers".

© 2021 Microchip Technology Inc.

29.5 Power, Clocks and Reset

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

29.5.1 POWER DOMAINS

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

29.5.2 CLOCK INPUTS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).
32KHz Core	This clock signal drives selected logic (e.g., counters).

29.5.3 RESETS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.

29.6 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
FAN_FAIL	The DRIVE_FAIL & FAN_SPIN bits in the Fan Status Register are logically ORed and routed to the FAIL_SPIN Interrupt
FAN_STALL	The FAN_STALL bit in the Fan Status Register is routed to the FAN_STALL Interrupt

29.7 Low Power Modes

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

29.8 Description

This section defines the functionality of the block.

29.8.1 GENERAL OPERATION

The RPM-PWM Interface is an RPM based Fan Control Algorithm that monitors the fan's speed and automatically adjusts the drive to maintain the desired fan speed. This RPM based Fan Control Algorithm controls a PWM output based on a tachometer input.

29.8.2 FAN CONTROL MODES OF OPERATION

The RPM-PWM Interface has two modes of operation for the PWM Fan Driver. They are:

1. Manual Mode - in this mode of operation, the user directly controls the fan drive setting. Updating the Fan Driver Setting Register (see Section 29.9.1, "Fan Setting Register") will update the fan drive based on the programmed ramp rate (default disabled).

- The Manual Mode is enabled by clearing the EN_ALGO bit in the Fan Configuration Register (see Section 29.9.2, "Fan Configuration Register").
- Whenever the Manual Mode is enabled the current drive settings will be changed to what was last used by the RPM control algorithm.
- Setting the drive value to 00h will disable the PWM Fan Driver.
- Changing the drive value from 00h will invoke the Spin Up Routine.

2. Using RPM based Fan Control Algorithm - in this mode of operation, the user determines a target tachometer reading and the drive setting is automatically updated to achieve this target speed.

© 2021 Microchip Technology Inc.

Manual Mode	Algorithm
Fan Driver Setting (read / write)	Fan Driver Setting (read only)
EDGES[1:0] (Fan Configuration)	EDGES[1:0] (Fan Configuration)
UPDATE[2:0] (Fan configuration)	UPDATE[2:0] (Fan configuration)
LEVEL (Spin Up Configuration)	LEVEL (Spin Up Configuration)
SPINUP_TIME[1:0] (Spin Up Configuration)	SPINUP_TIME[1:0] (Spin Up Configuration)
Fan Step	Fan Step
-	Fan Minimum Drive
Valid TACH Count	Valid TACH Count
-	TACH Target
TACH Reading	TACH Reading
RANGE[2:0] (Fan Configuration 2)	RANGE[2:0] (Fan Configuration 2)
-	DRIVE_FAIL_CNT[2:0] (Spin Up Config) and Drive Fail Band

29.8.3 RPM BASED FAN CONTROL ALGORITHM

The RPM-PWM Interface includes an RPM based Fan Control Algorithm.

The fan control algorithm uses Proportional, Integral, and Derivative terms to automatically approach and maintain the system's desired fan speed to an accuracy directly proportional to the accuracy of the clock source. Figure 29-2, "RPM based Fan Control Algorithm" shows a simple flow diagram of the RPM based Fan Control Algorithm operation.

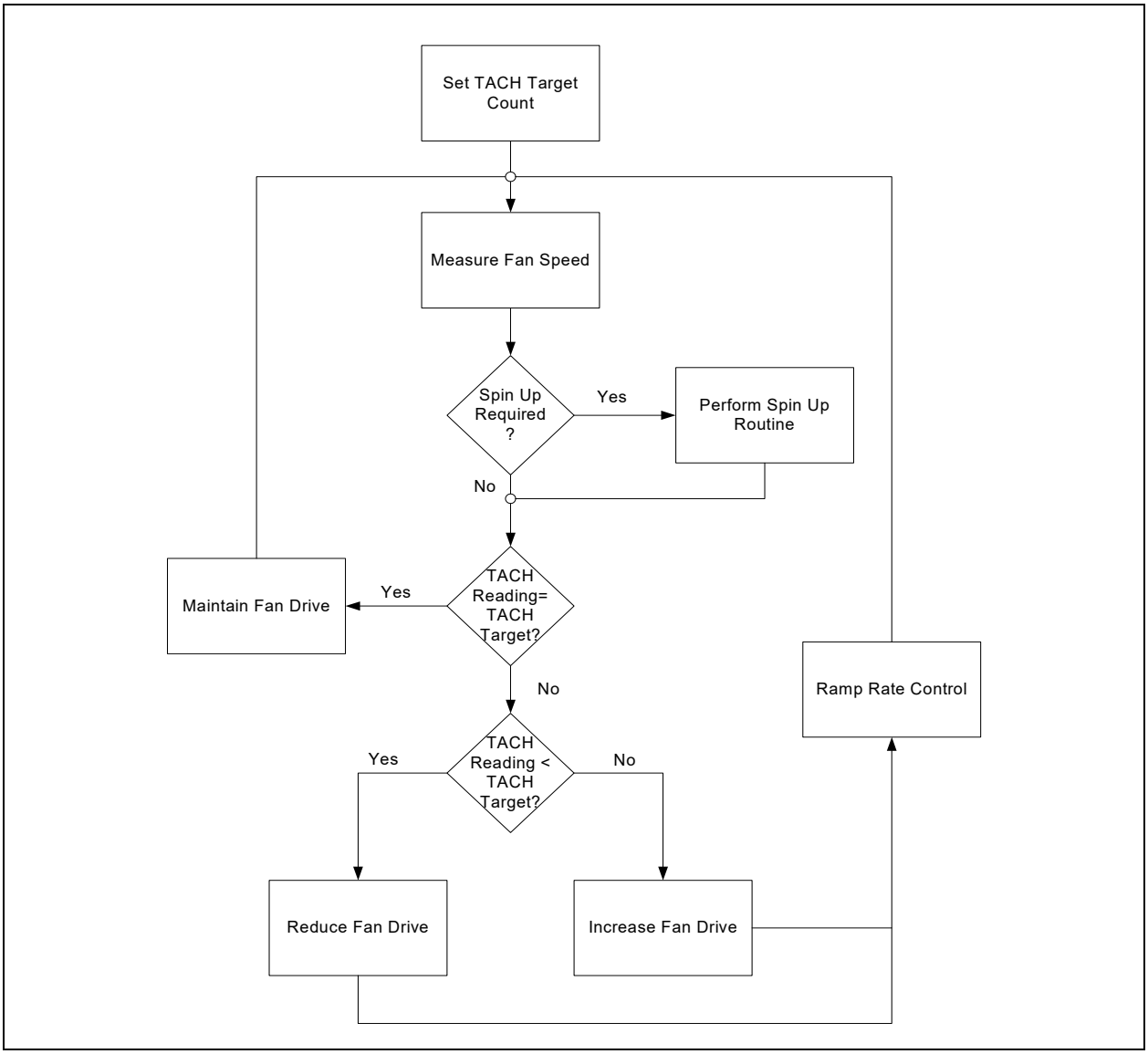
The desired tachometer count is set by the user inputting the desired number of 32.768KHz cycles that occur per fan revolution. The user may change the target count at any time. The user may also set the target count to FFh in order to disable the fan driver.

For example, if a desired RPM rate for a 2-pole fan is 3000 RPMs, the user would input the hexadecimal equivalent of 1312d (52_00h in the TACH Target Registers). This number represents the number of 32.768KHz cycles that would occur during the time it takes the fan to complete a single revolution when it is spinning at 3000RPMs (see Section 29.9.10, "TACH Target Register" and Section 29.9.11, "TACH Reading Register").

The RPM-PWM Interface's RPM based Fan Control Algorithm has programmable configuration settings for parameters such as ramp-rate control and spin up conditions. The fan driver automatically detects and attempts to alleviate a stalled/stuck fan condition while also asserting the interrupt signal. The RPM-PWM Interface works with fans that operate up to 16,000 RPMs and provide a valid tachometer signal.

© 2021 Microchip Technology Inc.

FIGURE 29-2: RPM BASED FAN CONTROL ALGORITHM



29.8.3.1 Programming the RPM Based Fan Control Algorithm

The RPM based Fan Control Algorithm powers-up disabled. The following registers control the algorithm. The RPM-PWM Interface fan control registers are pre-loaded with defaults that will work for a wide variety of fans so only the TACH Target Register is required to set a fan speed. The other fan control registers can be used to fine-tune the algorithm behavior based on application requirements.

1. Set the Valid TACH Count Register to the minimum tachometer count that indicates the fan is spinning.
2. Set the Spin Up Configuration Register to the spin up level and Spin Time desired.
3. Set the Fan Step Register to the desired step size.
4. Set the Fan Minimum Drive Register to the minimum drive value that will maintain fan operation.
5. Set the Update Time, and Edges options in the Fan Configuration Register.
6. Set the TACH Target Register to the desired tachometer count.
7. Enable the RPM based Fan Control Algorithm by setting the EN_ALGO bit.

29.8.3.2 Tachometer Measurement

In both modes of operation, the tachometer measurement operates independently of the mode of operation of the fan driver and RPM based Fan Speed Control algorithm. Any tachometer reading that is higher than the Valid TACH Count (see Section 29.9.8, "Valid TACH Count Register") will flag a stalled fan and trigger an interrupt.

When measuring the tachometer, the fan must provide a valid tachometer signal at all times to ensure proper operation. The tachometer measurement circuitry is programmable to detect the fan speed of a variety of fan configurations and architectures including 1-pole, 2-pole (default), 3-pole, and 4-pole fans.

Note:	The tachometer measurement works independently of the drive settings. If the device is put into manual mode and the fan drive is set at a level that is lower than the fan can operate (including zero drive), the tachometer measurement may signal a Stalled Fan condition and assert an interrupt.

STALLED FAN

If the TACH Reading Register exceeds the user-programmable Valid TACH Count setting, it will flag the fan as stalled and trigger an interrupt. If the RPM based Fan Control Algorithm is enabled, the algorithm will automatically attempt to restart the fan until it detects a valid tachometer level or is disabled.

The FAN_STALL Status bit indicates that a stalled fan was detected. This bit is checked conditionally depending on the mode of operation.

- Whenever the Manual Mode is enabled or whenever the drive value is changed from 00h, the FAN_STALL interrupt will be masked for the duration of the programmed Spin Up Time (see Section 29.9.5, "Fan Spin Up Configuration Register") to allow the fan an opportunity to reach a valid speed without generating unnecessary interrupts.
- In Manual Mode, whenever the TACH Reading Register exceeds the Valid TACH Count Register setting, the FAN_STALL status bit will be set.
- When the RPM based Fan Control Algorithm, the stalled fan condition is checked whenever the Update Time is met and the fan drive setting is updated. It is not a continuous check.

29.8.3.3 Spin Up Routine

The RPM-PWM Interface also contains programmable circuitry to control the spin up behavior of the fan driver to ensure proper fan operation. The Spin Up Routine is initiated under the following conditions:

- The TACH Target High Byte Register value changes from a value of FFh to a value that is less than the Valid TACH Count (see Section 29.9.8, "Valid TACH Count Register").
- The RPM based Fan Control Algorithm's measured tachometer reading is greater than the Valid TACH Count.
- When in Manual Mode, the Drive Setting changes from a value of 00h.

When the Spin Up Routine is operating, the fan driver is set to full scale for one quarter of the total user defined spin up time. For the remaining spin up time, the fan driver output is set a a user defined level (30% to 65% drive).

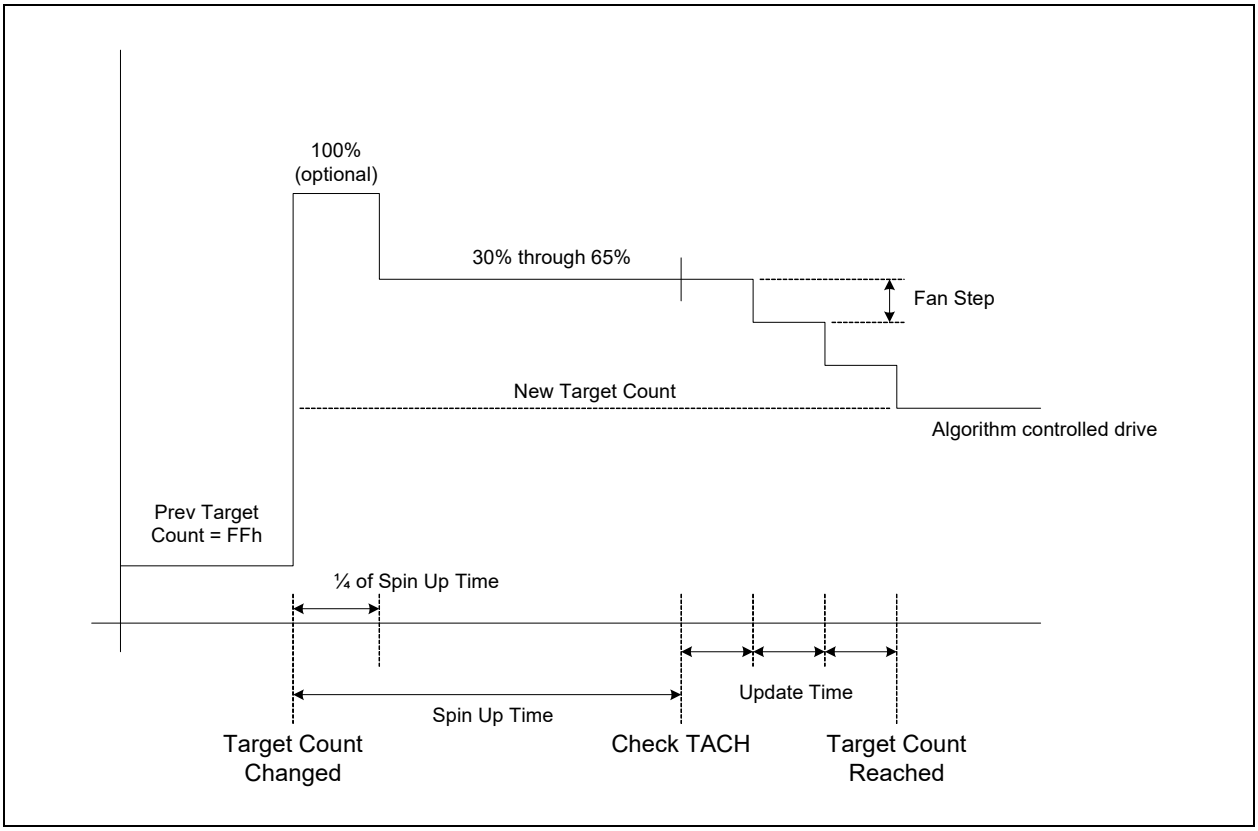
After the Spin Up Routine has finished, the RPM-PWM Interface measures the tachometer. If the measured tachometer reading is higher than the Valid TACH Count Register setting, the FAN_SPIN status bit is set and the Spin Up Routine will automatically attempt to restart the fan.

Note:	When the device is operating in manual mode, the FAN_SPIN status bit may be set if the fan drive is set at a level that is lower than the fan can operate (excluding zero drive which disables the fan driver). If the FAN_SPIN interrupt is unmasked, this condition will trigger an errant interrupt.

Figure 29-3, "Spin Up Routine" shows an example of the Spin Up Routine in response to a programmed fan speed change based on the first condition above.

© 2021 Microchip Technology Inc.

FIGURE 29-3: SPIN UP ROUTINE



29.8.4 PWM DRIVER

The RPM-PWM Interface contains an optional, programmable 10-bit PWM driver which can serve as part of the RPM based Fan Speed Control Algorithm or in Manual Mode.

When enabled, the PWM driver can operate in four programmable frequency bands. The lower frequency bands offer frequencies in the range of 9.5Hz to 4.8kHz while the higher frequency options offer frequencies of 21Hz or 25.2kHz.

The highest frequency available, 25.2KHz, operates in 8-bit resolution. All other PWM frequencies operate in 10-bit resolution.

29.8.5 FAN SETTING

The Fan Setting Registers are used to control the output of the Fan Driver. The driver setting operates independently of the Polarity bit for the PWM output. That is, a setting of 0000h will mean that the fan drive is at minimum drive while a value of FFC0h will mean that the fan drive is at maximum drive.

If the Spin Up Routine is invoked, reading from the registers will return the current fan drive setting that is being used by the Spin Up Routine instead of what was previously written into these registers.

The Fan Driver Setting Registers, when the RPM based Fan Control Algorithm is enabled, are read only. Writing to the register will have no effect and the data will not be stored. Reading from the register will always return the current fan drive setting.

If the INT_PWRGD pin is de-asserted, the Fan Driver Setting Register will be made read only. Writing to the register will have no effect and reading from the register will return 0000h.

When the RPM based Fan Control Algorithm is disabled, the current fan drive setting that was last used by the algorithm is retained and will be used.

If the Fan Driver Setting Register is set to a value of 0000h, all tachometer related status bits will be masked until the setting is changed. Likewise, the FAN_SHORT bit will be cleared and masked until the setting is changed.

The contents of the register represent the weighting of each bit in determining the final duty cycle. The output drive for a PWM output is given by the following equation:

- Drive = (FAN_SETTING VALUE/1023) x 100%.

The PWM Divide Register determines the final PWM frequency. The base frequency set by the PWM_BASE[1:0] bits is divided by the decimal equivalent of the register settings.

The final PWM frequency is derived as the base frequency divided by the value of this register as shown in the equation below:

- PWM_Frequency = base_clk / PWM_D

Where:

- base_clk = The base frequency set by the PWMx_CFG[1:0] bits
- PWM_D = the divide setting set by the PWM Divide Register.

29.8.6 ALERTS AND LIMITS

Figure 29-4, "Interrupt Flow" shows the interactions of the interrupts for fan events.

If the Fan Driver detects a drive fail, spin-up or stall event, the interrupt signal will be asserted (if enabled).

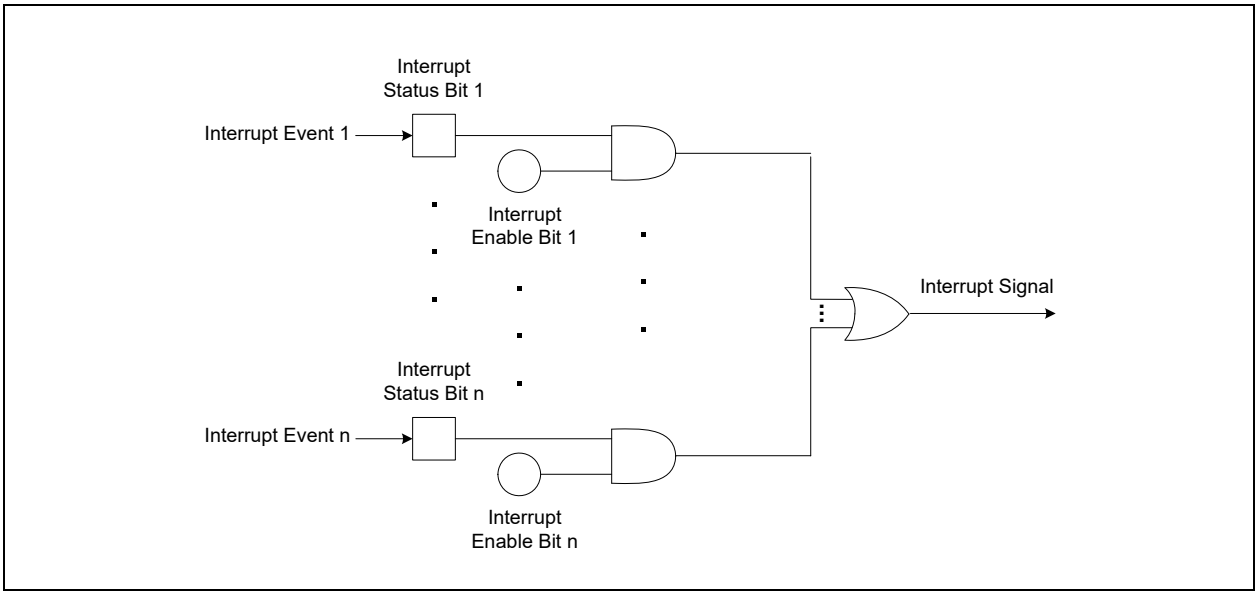
All of these interrupts can be masked from asserting the interrupt signal individually. If any bit of either Status register is set, the interrupt signal will be asserted provided that the corresponding interrupt enable bit is set accordingly.

The Status register will be updated due to an active event, regardless of the setting of the individual enable bits. Once a status bit has been set, it will remain set until the Status register bit is written to 1 (and the error condition has been removed).

If the interrupt signal is asserted, it will be cleared immediately if either the status or enable bit is cleared.

See Section 29.6, "Interrupts".

FIGURE 29-4: INTERRUPT FLOW



© 2021 Microchip Technology Inc.

29.9 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 RPM-PWM Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 29-1: REGISTER SUMMARY

Offset	Register Name
00h	Fan Setting
02h	Fan Configuration Register
04h	PWM Divide Register
05h	Gain Register
06h	Fan Spin Up Configuration Register
07h	Fan Step Register
08h	Fan Minimum Drive Register
09h	Valid TACH Count Register
0Ah	Fan Drive Fail Band Register
0Ch	TACH Target Register
0Eh	TACH Reading Register
10h	PWM Driver Base Frequency Register
11h	Fan Status Register

29.9.1 FAN SETTING REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:6	FAN_SETTING The Fan Driver Setting used to control the output of the Fan Driver.	R/W	00h	RESET_SYS
5:0	Reserved	R	-	-

29.9.2 FAN CONFIGURATION REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	EN_RRC Enables the ramp rate control circuitry during the Manual Mode of operation. 1=The ramp rate control circuitry for the Manual Mode of operation is enabled. The PWM setting will follow the ramp rate controls as determined by the Fan Step and Update Time settings. The maximum PWM step is capped at the Fan Step setting and is updated based on the Update Time as given by the field UPDATE. 0=The ramp rate control circuitry for the Manual Mode of operation is disabled. When the Fan Drive Setting values are changed and the RPM based Fan Control Algorithm is disabled, the fan driver will be set to the new setting immediately.	R/W	0b	RESET_SYS
14	DIS_GLITCH Disables the low pass glitch filter that removes high frequency noise injected on the TACH pin. 1=The glitch filter is disabled 0=The glitch filter is enabled	R/W	0b	RESET_SYS
13:12	DER_OPT Control some of the advanced options that affect the derivative portion of the RPM based fan control algorithm as shown in Table 29-3, "Derivative Options". These bits only apply if the Fan Speed Control Algorithm is used.	R/W	3h	RESET_SYS
11:10	ERR_RNG Control some of the advanced options that affect the error window. When the measured fan speed is within the programmed error window around the target speed, the fan drive setting is not updated. These bits only apply if the Fan Speed Control Algorithm is used. 3=200 RPM 2=100 RPM 1=50 RPM 0=0 RPM	R/W	1h	RESET_SYS
9	POLARITY Determines the polarity of the PWM driver. This does NOT affect the drive setting registers. A setting of 0% drive will still correspond to 0% drive independent of the polarity. 1=The Polarity of the PWM driver is inverted. A drive setting of 00h will cause the output to be set at 100% duty cycle and a drive setting of FFh will cause the output to be set at 0% duty cycle. 0=The Polarity of the PWM driver is normal. A drive setting of 00h will cause the output to be set at 0% duty cycle and a drive setting of FFh will cause the output to be set at 100% duty cycle.	R/W	0h	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
8	Reserved	R	-	-
7	EN_ALGO Enables the RPM based Fan Control Algorithm. 1=The control circuitry is enabled and the Fan Driver output will be automatically updated to maintain the programmed fan speed as indicated by the TACH Target Register. 0=The control circuitry is disabled and the fan driver output is determined by the Fan Driver Setting Register.	R/W	0b	RESET_SYS
6:5	RANGE Adjusts the range of reported and programmed tachometer reading values. The RANGE bits determine the weighting of all TACH values (including the Valid TACH Count, TACH Target, and TACH reading). 3=Reported Minimum RPM: 4000. Tach Count Multiplier: 8 2=Reported Minimum RPM: 2000. Tach Count Multiplier: 4 1=Reported Minimum RPM: 1000. Tach Count Multiplier: 2 0=Reported Minimum RPM: 500. Tach Count Multiplier: 1	R/W	1h	RESET_SYS

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4:3	EDGES Determines the minimum number of edges that must be detected on the TACH signal to determine a single rotation. A typical fan measured 5 edges (for a 2-pole fan). Increasing the number of edges measured with respect to the number of poles of the fan will cause the TACH Reading registers to indicate a fan speed that is higher or lower than the actual speed. In order for the FSC Algorithm to operate correctly, the TACH Target must be updated by the user to accommodate this shift. The Effective Tach Multiplier shown in Table 29-2, "Minimum Edges for Fan Rotation" is used as a direct multiplier term that is applied to the Actual RPM to achieve the Reported RPM. It should only be applied if the number of edges measured does not match the number of edges expected based on the number of poles of the fan (which is fixed for any given fan). Contact Microchip for recommended settings when using fans with more or less than 2 poles.	R/W	1h	RESET_SYS
2:0	UPDATE Determines the base time between fan driver updates. The Update Time, along with the Fan Step Register, is used to control the ramp rate of the drive response to provide a cleaner transition of the actual fan operation as the desired fan speed changes. 7=1600ms 6=1200ms 5=800ms 4=500ms 3=400ms 2=300ms 1=200ms 0=100ms **Note:** This ramp rate control applies for all changes to the active PWM output including when the RPM based Fan Speed Control Algorithm is disabled.	R/W	3h	RESET_SYS

TABLE 29-2: MINIMUM EDGES FOR FAN ROTATION

Edges	Minimum TACH Edges	Number of Fan Poles	Effective TACH Multiplier (Based on 2 Pole Fans) If Edges Changed
0h	3	1	0.5
1h	5	2 (default)	1
2h	7	3	1.5
3h	9	4	2

© 2021 Microchip Technology Inc.

TABLE 29-3: DERIVATIVE OPTIONS

DER_OPT	Operation	Note (see Section 29.9.6, "Fan Step Register")
0	No derivative options used	PWM steps are limited to the maximum PWM drive step value in Fan Step Register
1	Basic derivative. The derivative of the error from the current drive setting and the target is added to the iterative PWM drive setting (in addition to proportional and integral terms)	PWM steps are limited to the maximum PWM drive step value in Fan Step Register
2	Step derivative. The derivative of the error from the current drive setting and the target is added to the iterative PWM drive setting and is not capped by the maximum PWM drive step. This allows for very fast response times	PWM steps are not limited to the maximum PWM drive step value in Fan Step Register (i.e., maximum fan step setting is ignored)
3	Both the basic derivative and the step derivative are used effectively causing the derivative term to have double the effect of the derivative term (default).	PWM steps are not limited to the maximum PWM drive step value in Fan Step Register (i.e., maximum fan step setting is ignored)

29.9.3 PWM DIVIDE REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	PWM_DIVIDE The PWM Divide value determines the final frequency of the PWM driver. The driver base frequency is divided by the PWM Divide value to determine the final frequency.	R/W	01h	RESET_SYS

29.9.4 GAIN REGISTER

The Gain Register stores the gain terms used by the proportional and integral portions of the RPM based Fan Control Algorithm. These terms will affect the FSC closed loop acquisition, overshoot, and settling as would be expected in a classic PID system.

This register only applies if the Fan Speed Control Algorithm is used.

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	R	-	-
5:4	GAIND The derivative gain term. Gain Factor: 3=8x 2=4x 1=2x 0=1x	R/W	2h	RESET_SYS

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3:2	GAINI The integral gain term. Gain Factor: 3=8x 2=4x 1=2x 0=1x	R/W	2h	RESET_SYS
1:0	GAINP The proportional gain term. Gain Factor: 3=8x 2=4x 1=2x 0=1x	R/W	2h	RESET_SYS

29.9.5 FAN SPIN UP CONFIGURATION REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	DRIVE_FAIL_CNT Determines how many update cycles are used for the Drive Fail detection function. This circuitry determines whether the fan can be driven to the desired Tach target. These settings only apply if the Fan Speed Control Algorithm is enabled. 3=Drive Fail detection circuitry will count for 64 update periods 2=Drive Fail detection circuitry will count for 32 update periods 1=Drive Fail detection circuitry will count for 16 update periods 0=Drive Fail detection circuitry is disabled	R/W	00b	RESET_SYS
5	NOKICK Determines if the Spin Up Routine will drive the fan to 100% duty cycle for 1/4 of the programmed spin up time before driving it at the programmed level. 1=The Spin Up Routine will not drive the PWM to 100%. It will set the drive at the programmed spin level for the entire duration of the programmed spin up time 0=The Spin Up Routine will drive the PWM to 100% for 1/4 of the programmed spin up time before reverting to the programmed spin level	R/W	0b	RESET_SYS

© 2021 Microchip Technology Inc.

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4:2	SPIN_LVL Determines the final drive level that is used by the Spin Up Routine. 7=65% 6=60% 5=55% 4=50% 3=45% 2=40% 1=35% 0=30%	R/W	6h	RESET_SYS
1:0	SPINUP_TIME Determines the maximum Spin Time that the Spin Up Routine will run for. If a valid tachometer measurement is not detected before the Spin Time has elapsed, an interrupt will be generated. When the RPM based Fan Control Algorithm is active, the fan driver will attempt to re-start the fan immediately after the end of the last spin up attempt. 3=2 seconds 2=1 second 1=500 ms 0=250 ms	R/W	1h	RESET_SYS

29.9.6 FAN STEP REGISTER

The Fan Step Register, along with the Update Time, controls the ramp rate of the fan driver response calculated by the RPM based Fan Control Algorithm for the Derivative Options field values of “00” and “01” in the Fan Configuration Register.

The value of the register represents the maximum step size the fan driver will take for each update.

When the maximum step size limitation is applied, if the necessary fan driver delta is larger than the Fan Step, it will be capped at the Fan Step setting and updated every Update Time ms.

The maximum step size is ignored for the Derivative Options field values of “10” and “11”.

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	FAN_STEP The Fan Step value represents the maximum step size the fan driver will take between update times. When the PWM_BASE frequency range field in the PWM Driver Base Frequency Register is set to the value 1, 2 or 3, this 8-bit field is added to the 10-bit PWM duty cycle, for a maximum step size of 25%. When the PWM_BASE field is set to 0, the PWM operates in an 8-bit mode. In 8-bit mode, this 8-bit field is added to the 8-bit duty cycle, for a maximum step size of 100%.	R/W	10h	RESET_SYS

29.9.7 FAN MINIMUM DRIVE REGISTER

the Fan Minimum Drive Register stores the minimum drive setting for the RPM based Fan Control Algorithm. The RPM based Fan Control Algorithm will not drive the fan at a level lower than the minimum drive unless the target Fan Speed is set at FFh (see "TACH Target Registers").

During normal operation, if the fan stops for any reason (including low drive), the RPM based Fan Control Algorithm will attempt to restart the fan. Setting the Fan Minimum Drive Registers to a setting that will maintain fan operation is a useful way to avoid potential fan oscillations as the control circuitry attempts to drive it at a level that cannot support fan operation.

These registers only apply if the Fan Speed Control Algorithm is used.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MIN_DRIVE The minimum drive setting.	R/W	66h	RESET_SYS

Note:	To ensure proper operation, the Fan Minimum Drive register must be set prior to setting the Tach Target High and Low Byte registers, and then the Tach Target registers can be subsequently updated. At a later time, if the Fan Minimum Drive register is changed to a value higher than current Fan value, the Tach Target registers must also be updated.

29.9.8 VALID TACH COUNT REGISTER

The Valid TACH Count Register stores the maximum TACH Reading Register value to indicate that the fan is spinning properly. The value is referenced at the end of the Spin Up Routine to determine if the fan has started operating and decide if the device needs to retry. See the equation in the TACH Reading Registers section for translating the RPM to a count.

If the TACH Reading Register value exceeds the Valid TACH Count Register (indicating that the Fan RPM is below the threshold set by this count), a stalled fan is detected. In this condition, the algorithm will automatically begin its Spin Up Routine.

Note:	The automatic invoking of the Spin Up Routine only applies if the Fan Speed Control Algorithm is used. If the FSC is disabled, then the device will only invoke the Spin Up Routine when the PWM setting changes from 00h.

If a TACH Target setting is set above the Valid TACH Count setting, that setting will be ignored and the algorithm will use the current fan drive setting.

These registers only apply if the Fan Speed Control Algorithm is used.

Offset	09h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	VALID_TACH_CNT The maximum TACH Reading Register value to indicate that the fan is spinning properly.	R/W	F5h	RESET_SYS

© 2021 Microchip Technology Inc.

29.9.9 FAN DRIVE FAIL BAND REGISTER

The Fan Drive Fail Band Registers store the number of Tach counts used by the Fan Drive Fail detection circuitry. This circuitry is activated when the fan drive setting high byte is at FFh. When it is enabled, the actual measured fan speed is compared against the target fan speed.

This circuitry is used to indicate that the target fan speed at full drive is higher than the fan is actually capable of reaching. If the measured fan speed does not exceed the target fan speed minus the Fan Drive Fail Band Register settings for a period of time longer than set by the DRIVE_FAIL_CNTx[1:0] bits in the Fan Spin Up Configuration Register, the DRIVE_FAIL status bit will be set and an interrupt generated.

These registers only apply if the Fan Speed Control Algorithm is used.

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:3	FAN_DRIVE_FAIL_BAND The number of Tach counts used by the Fan Drive Fail detection circuitry	R	0h	RESET_SYS
2:0	Reserved	R	-	-

29.9.10 TACH TARGET REGISTER

The TACH Target Registers hold the target tachometer value that is maintained for the RPM based Fan Control Algorithm.

If the algorithm is enabled, setting the TACH Target Register High Byte to FFh will disable the fan driver (or set the PWM duty cycle to 0%). Setting the TACH Target to any other value (from a setting of FFh) will cause the algorithm to invoke the Spin Up Routine after which it will function normally.

These registers only apply if the Fan Speed Control Algorithm is used.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:3	TACH_TARGET The target tachometer value.	R	-	RESET_SYS
2:0	Reserved	R	-	-

29.9.11 TACH READING REGISTER

The TACH Reading Registers' contents describe the current tachometer reading for the fan. By default, the data represents the fan speed as the number of 32.768kHz clock periods that occur for a single revolution of the fan.

The Equation below shows the detailed conversion from tachometer measurement (COUNT) to RPM.

$$RPM = \frac{1}{Poles} \times \frac{(n-1)}{COUNT \times \frac{1}{m}} \times f_{TACH} \times 60$$

where:

- *Poles* = number of poles of the fan (typically 2)
- f_{TACH} = the frequency of the tachometer measurement clock
- *n* = number of edges measured (typically 5 for a 2 pole fan)
- *m* = the multiplier defined by the RANGE bits
- *COUNT* = TACH Reading Register value (in decimal)

The following equation shows the simplified translation of the TACH Reading Register count to RPM assuming a 2-pole fan, measuring 5 edges, with a frequency of 32.768kHz.

$$RPM = \frac{3932160 \times m}{COUNT}$$

Offset	0Eh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:3	TACH_READING The current tachometer reading value.	R	-	RESET_SYS
2:0	Reserved	R	-	-

29.9.12 PWM DRIVER BASE FREQUENCY REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	R	-	-
1:0	PWM_BASE Determines the frequency range of the PWM fan driver (when enabled). PWM resolution is 10-bit, except when this field is set to '0b', when it is 8-bit. 3=2.34KHz 2=4.67KHz 1=23.4KHz 0=26.8KHz	R/W	00b	RESET_SYS

© 2021 Microchip Technology Inc.

29.9.13 FAN STATUS REGISTER

Offset	11h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	R	-	-
5	DRIVE_FAIL The bit Indicates that the RPM-based Fan Speed Control Algorithm cannot drive the Fan to the desired target setting at maximum drive. 1=The RPM-based Fan Speed Control Algorithm cannot drive Fan to the desired target setting at maximum drive. 0=The RPM-based Fan Speed Control Algorithm can drive Fan to the desired target setting.	R/WC	0b	RESET_SYS
4:2	Reserved	R	-	-
1	FAN_SPIN The bit Indicates that the Spin up Routine for the Fan could not detect a valid tachometer reading within its maximum time window. 1=The Spin up Routine for the Fan could not detect a valid tachometer reading within its maximum time window. 0=The Spin up Routine for the Fan detected a valid tachometer reading within its maximum time window.	R/WC	0b	RESET_SYS
0	FAN_STALL The bit Indicates that the tachometer measurement on the Fan detects a stalled fan. 1=Stalled fan not detected 0=Stalled fan not detected	R/WC	0b	RESET_SYS

29.10 Usage Models

The example below explains the usage/ register programming of this block for a 2 pole Fan with RPM value less than 500.

Example

Most fans are two pole and thus require 5 edges to calculate one revolution. In cases where you need a minimum fan speed of less than 500 RPM, 3 edges can be used to measure half of a revolution. This allows for a lower fan speed before a Tach reaches its end count.

The equation for determining RPM as function of Tach count when the proper edge selection is done is given below:

RPM = (392160 * M) / count

When 3 edges is chosen instead of 5 edge for a 2 pole fan, the RPM equation is as follows:

RPM = (392160 * M) / (count *2)

When 3 edges instead of 5 edges are used to determine fan speed, the modified equation must be used for calculating and programming TACH Target Register and Valid TACH Count Register or reading from TACH Reading Register to determine appropriate fan speed. This would require software that use the TACH Reading Register to use the new equation mentioned above to display the fan speed properly.

Note:	The Valid TACH Count Register is a 8 bit register instead of a 13 bit. These 8 bits should be treated as the upper 8 bits of a 13 bit count value.

© 2021 Microchip Technology Inc.

30.0 QUAD SPI MASTER CONTROLLER

30.1 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.

30.2 References

No references have been cited for this feature.

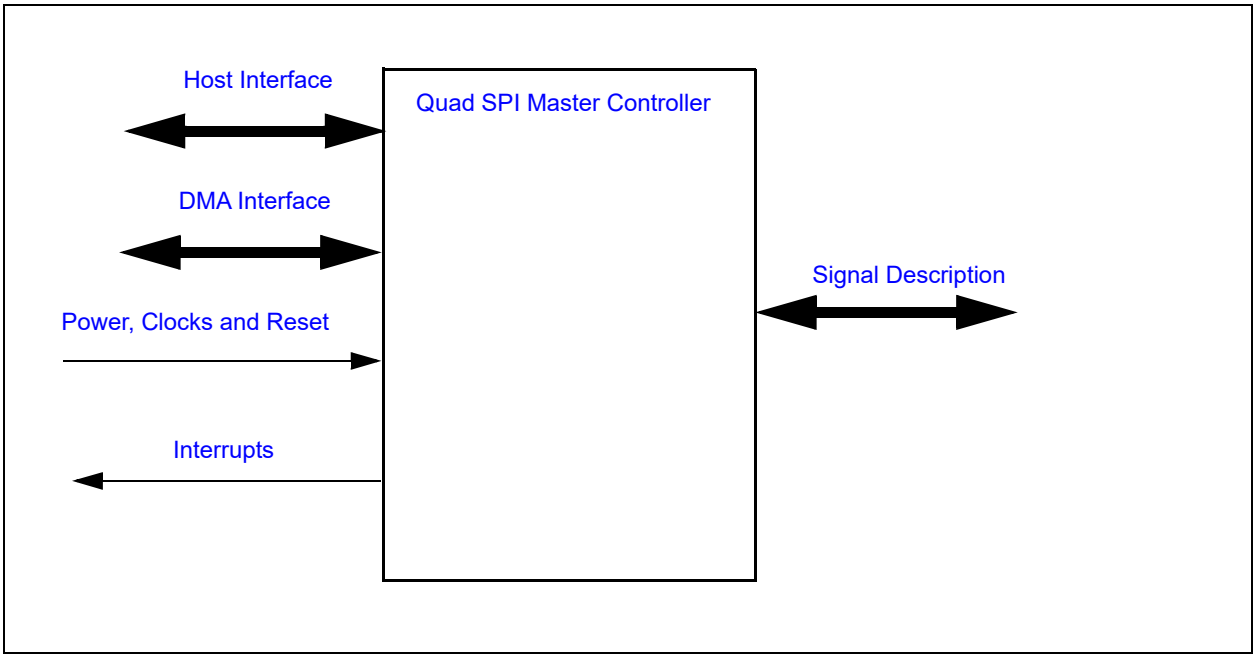
30.3 Terminology

No terminology for this block.

30.4 Interface

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

FIGURE 30-1: I/O DIAGRAM OF BLOCK



30.5 Signal Description

TABLE 30-1: 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

© 2021 Microchip Technology Inc.

TABLE 30-1: 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.

30.6 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".

30.7 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 8.0, "Internal DMA Controller".

30.8 Power, Clocks and Reset

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

30.8.1 POWER

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

30.8.2 CLOCKS

Name	Description
96 MHz	This is a clock source for the SPI clock generator.

30.8.3 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.

30.9 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.

30.10 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’.

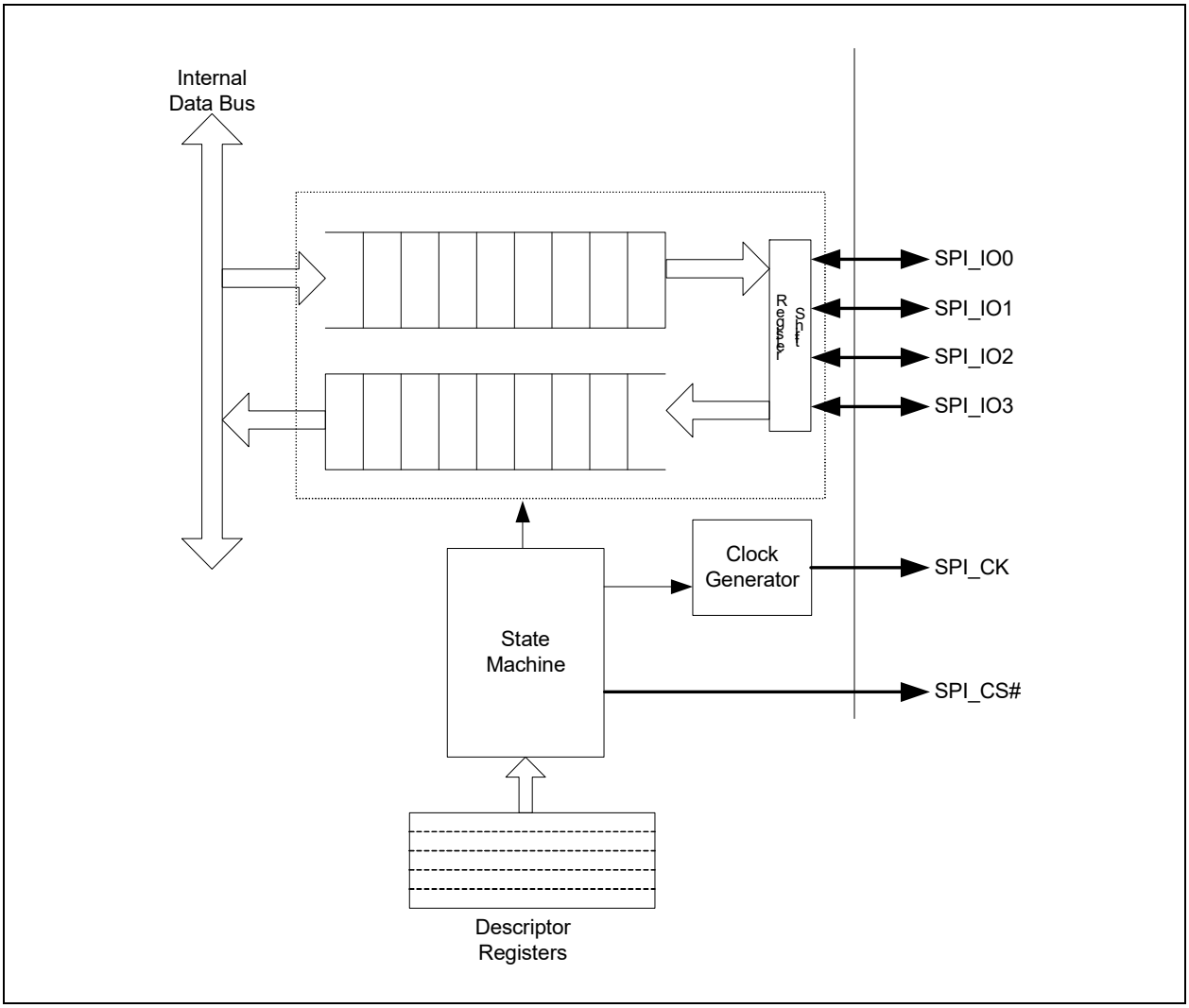
If the QMSPI SLEEP_ENABLE input is asserted, writes to the START bit are ignored and the Quad SPI Master Controller will remain in its lowest power state.

© 2021 Microchip Technology Inc.

30.11 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 30-2: QUAD MASTER SPI BLOCK DIAGRAM



30.11.1 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

30.11.2 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
- Local DMA. Supports local Rx and Tx DMA channels to transfer data at high rates.

30.11.3 SPI CLOCK

The SPI output clock is derived from the 96 MHz, divided by a value programmed in the CLOCK_DIVIDE field of the QMSPI Mode Register. Sample frequencies are shown in the following table:

TABLE 30-2: EXAMPLE SPI FREQUENCIES

CLOCK_DIVIDE	**SPI Clock Frequency**
0	375 KHz
1	96 MHz
2	48 MHz
3	36 MHz
6	16 MHz
48	2 MHz
128	750 KHz
255	376.5 KHz

30.11.4 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

© 2021 Microchip Technology Inc.

30.12 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 30-3: 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
C0	QMSPI Mode Alternate1 Register
D0	Test
D4	QMSPI Taps Adjustment Register
D8	Test
100h	QMSPI Descriptor Local DMA Rx Enable Register
104h	QMSPI Descriptor Local DMA Tx Enable Register
110h	QMSPI Local DMA Rx Control Channel 0 Register
114h	QMSPI Local DMA Rx Start Address Channel 0 Register
118h	QMSPI Local DMA Rx Length Channel 0 Register
11Ch	Reserved
120h	QMSPI Local DMA Rx Control Channel 1 Register

© 2021 Microchip Technology Inc.

TABLE 30-3: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
124h	QMSPI Local DMA Rx Start Address Channel 1 Register
128h	QMSPI Local DMA Rx Length Channel 1 Register
12Ch	Reserved
120h	QMSPI Local DMA Rx Control Channel 2 Register
124h	QMSPI Local DMA Rx Start Address Channel 2 Register
128h	QMSPI Local DMA Rx Length Channel 2 Register
12Ch	Reserved
140h	QMSPI Local DMA Tx Control Channel 0 Register
144h	QMSPI Local DMA Tx Start Address Channel 0 Register
148h	QMSPI Local DMA Tx Length Channel 0 Register
14Ch	Reserved
150h	QMSPI Local DMA Tx Control Channel 1 Register
154h	QMSPI Local DMA Tx Start Address Channel 1 Register
158h	QMSPI Local DMA Tx Length Channel 1 Register
15Ch	Reserved
160h	QMSPI Local DMA Tx Control Channel 2 Register
164h	QMSPI Local DMA Tx Start Address Channel 2 Register
168h	QMSPI Local DMA Tx Length Channel 2 Register
16Ch	Reserved

30.12.1 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 30-2, "Example SPI Frequencies" for examples.	R/W	0h	RESET
15:14	Reserved	RES	-	-
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	-	-

© 2021 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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
7:5	Reserved	RES	-	-
4	Local DMA Tx Enable This enables the Local DMA usage (instead of the Central DMA) when the Control register enables the DMA. 0 = Central DMA for Tx DMA Enable 1 = Local DMA for Tx DMA Enable	R/W	0h	RESET
3	Local DMA Rx Enable This enables the Local DMA usage (instead of the Central DMA) when the Control register enables the DMA. 0 = Central DMA for Rx DMA Enable 1 = Local DMA for Rx DMA Enable	R/W	0h	RESET

© 2021 Microchip Technology Inc.

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

© 2021 Microchip Technology Inc.

30.12.2 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 When the local DMA is in use: This selects what channel of the local Rx DMA is selected. If 0, DMA is disabled. If 1-3 local Rx DMA channel 1-3 is selected.	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 When the local DMA is in use: This selects what channel of the local Tx DMA is selected. If 0, DMA is disabled. If 1-3 local Tx DMA channel 1-3 is selected.	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

30.12.3 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

© 2021 Microchip Technology Inc.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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. After the transfer has stopped, the controller will de-assert chip-select to terminate the transfer over the SPI interface This bit must not be set to '1' if the field START in this register is set to '1'.	W	0h	RESET
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

30.12.4 QMSPI INTERFACE CONTROL REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
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
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

30.12.5 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
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

© 2021 Microchip Technology Inc.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
8	TRANSMIT_BUFFER_FULL 1=The Transmit Buffer is full 0=The Transmit Buffer is not full	R	0h	RESET
7	Reserved	RES	-	-
6	LOCAL_DMA_TX_ERROR 1=Error during transfer 0= No Error	R	1h	RESET
5	LOCAL_DMA_RX_ERROR 1=Error during transfer 0=No Error	R	0h	RESET
4	PROGRAMMING_ERROR This bit if a programming error is detected. Programming errors are listed in Section 30.11.4, "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

30.12.6 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

30.12.7 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	-	-

© 2021 Microchip Technology Inc.

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	Reserved	RES	-	-
6	LOCAL_DMA_TX_ERR_ENABLE 1=Enable an interrupt if is LOCAL_DMA_TX_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
5	LOCAL_DMA_RX_ERR_ENABLE 1=Enable an interrupt if LOCAL_DMA_RX_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
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

30.12.8 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

30.12.9 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

30.12.10 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

© 2021 Microchip Technology Inc.

30.12.11 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
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

30.12.12 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

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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_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 **Note:** When the local DMA is in use: RX_DMA_ENABLE selects what channel of the local Rx DMA is selected. If 0, DMA is disabled. If 1 to 3 local Rx DMA channel 1 to 3 is selected.	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 **Note:** When the local DMA is in use: TX_DMA_ENABLE selects what channel of the local Tx DMA is selected. If 0, DMA is disabled. If 1 to 3 local Tx DMA channel 1 to 3 is selected.	R/W	0h	RESET

© 2021 Microchip Technology Inc.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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
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

30.12.13 QMSPI DESCRIPTION BUFFER 1 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.14 QMSPI DESCRIPTION BUFFER 2 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.15 QMSPI DESCRIPTION BUFFER 3 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.16 QMSPI DESCRIPTION BUFFER 4 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.17 QMSPI DESCRIPTION BUFFER 5 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.18 QMSPI DESCRIPTION BUFFER 6 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.19 QMSPI DESCRIPTION BUFFER 7 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.20 QMSPI DESCRIPTION BUFFER 8 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.21 QMSPI DESCRIPTION BUFFER 9 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.22 QMSPI DESCRIPTION BUFFER 10 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.23 QMSPI DESCRIPTION BUFFER 11 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.24 QMSPI DESCRIPTION BUFFER 12 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.25 QMSPI DESCRIPTION BUFFER 13 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.26 QMSPI DESCRIPTION BUFFER 14 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.27 QMSPI DESCRIPTION BUFFER 15 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

30.12.28 QMSPI MODE ALTERNATE1 REGISTER

Offset	C0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Chip Select 1 Alternate Clock Divide The SPI clock divide in number of system clocks when CS1 is in use and CS1 Alt Mode Enable is set.	R/W	0h	RESET
15:1	Reserved	RES	-	-
0	Chip Select 1 Alternate Mode Enable Enable the CS1 Clock Divide to be active if CS1 is the interface in use.	R/W	0h	RESET

30.12.29 QMSPI TAPS ADJUSTMENT REGISTER

Offset	D4h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:8	Select Control Tap Adjustment This is a signed value used to come up with the final value for the delay. This is used to adjust the auto-H/W trim if needed.	R/W	0h	RESET
7:0	Select SCK Tap Adjustment This is a signed value used to come up with the final value for the delay. This is used to adjust the auto-H/W trim if needed.	R/W	0h	RESET

© 2021 Microchip Technology Inc.

30.12.30 QMSPI DESCRIPTOR LOCAL DMA RX ENABLE REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	Local DMA Descriptor Rx Enable This enables the Local DMA usage (instead of the Central DMA) when the Descriptor Buffer register enables the DMA. Bit 0 is associated with Description Buffer[0] while bit 15 is associated with Description Buffer [15].	R/W	0h	RESET

30.12.31 QMSPI DESCRIPTOR LOCAL DMA TX ENABLE REGISTER

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	Local DMA Descriptor Tx Enable This enables the Local DMA usage (instead of the Central DMA) when the Descriptor Buffer register enables the DMA. Bit 0 is associated with Description Buffer[0] while bit 15 is associated with Description Buffer [15].	R/W	0h	RESET

30.12.32 QMSPI LOCAL DMA RX CONTROL CHANNEL 0 REGISTER

Offset	110h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	Local DMA Rx Increment Address Enable When set, the DMA Channel's Start Address will increment on every access. If not set the address will not increment; so it can be targeted at a FIFO style memory. 0=On Access: Start Address does not increment. 1=On Access: Start Address increments.	R/W	0h	RESET
5:4	Local DMA Rx Access Size Selects the AHB Access Size. 0=1 Byte 1=2 Bytes 2=4 Bytes	R/W	0h	RESET

Offset	110h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	Local DMA Rx Override Length This will override the length field to the QMSPI protocol FSM with the length programmed into the Local DMA. Do not have both Tx and Rx Local DMA's enabled with different lengths. This is a mis-programming case and will flag an error interrupt and abort the transfer. 0=Normal Length is used. 1=Length of transfer uses the DMA length rather than the standard control register length.	R/W	0h	RESET
2	Local DMA Rx Restart Address Enable When set, the DMA Channel's Start Address will reset to its initial value upon completion. This facilitates DMA Channel re-use without reprogramming. If this is not set, then the Start Address will be the last address accessed + transfer size upon completion. 0=On Completion: Start Address is last address accessed + transfer size. 1=On Completion: Start Address is reset to the initially programmed Start Address.	R/W	0h	RESET
1	Local DMA Rx Restart Enable This sets the DMA Channel to re-enable itself after a completion so the next DMA transfer can occur without requiring manual re-programming of the DMA Channel. 0=On Completion: DMA is disabled and needs to be restarted. 1=On Completion: DMA is re-enabled.	R/W	0h	RESET
0	Local DMA Rx Channel Enable This states that the DMA is programmed and ready to run. While this is cleared the QMSPI will be stalled, waiting for the DMA to being transferring, once the local FIFO is full. This is cleared by H/W once a transfer is completed. It can be re-set by H/W if Local DMA Restart is enabled. 0=The Local DMA Channel will not run. 1=The Local DMA Channel will run once the transfer requests this to function.	R/W	0h	RESET

30.12.33 QMSPI LOCAL DMA RX START ADDRESS CHANNEL 0 REGISTER

Offset	114h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Local DMA Start Address This enables the Local DMA usage (instead of the Central DMA) when the Descriptor Buffer register enables the DMA. Bit 0 is associated with Description Buffer[0] while bit 15 is associated with Description Buffer [15].	R/W	0h	RESET

© 2021 Microchip Technology Inc.

30.12.34 QMSPI LOCAL DMA RX LENGTH CHANNEL 0 REGISTER

Offset	118h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Local DMA Length Address This is the maximum Length of the transfer in Bytes that the DMA Channel will allow access to. Once this length is reached the DMA Channel will terminate any further accesses, like the Central DMA does. This length can be used as a Byte Length to the QMSPI FSM's in the override mode.	R/W	0h	RESET

30.12.35 QMSPI LOCAL DMA RX CONTROL CHANNEL 1 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Rx Control Channel 0 Register.

30.12.36 QMSPI LOCAL DMA RX START ADDRESS CHANNEL 1 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Rx Start Address Channel 0 Register.

30.12.37 QMSPI LOCAL DMA RX LENGTH CHANNEL 1 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Rx Length Channel 0 Register.

30.12.38 QMSPI LOCAL DMA RX CONTROL CHANNEL 2 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Rx Control Channel 0 Register.

30.12.39 QMSPI LOCAL DMA RX START ADDRESS CHANNEL 2 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Rx Start Address Channel 0 Register.

30.12.40 QMSPI LOCAL DMA RX LENGTH CHANNEL 2 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Rx Length Channel 0 Register.

30.12.41 QMSPI LOCAL DMA TX CONTROL CHANNEL 0 REGISTER

Offset	140h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	Local DMA Tx Increment Address Enable When set, the DMA Channel's Start Address will increment on every access. If not set the address will not increment; so it can be targeted at a FIFO style memory. 0=On Access: Start Address does not increment. 1=On Access: Start Address increments.	R/W	0h	RESET
5:4	Local DMA Tx Access Size Selects the AHB Access Size. 0=1 Byte 1=2 Bytes 2=4 Bytes	R/W	0h	RESET

Offset	140h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	Local DMA Tx Override Length This will override the length field to the QMSPI protocol FSM with the length programmed into the Local DMA. Do not have both Tx and Rx Local DMA's enabled with different lengths. This is a mis-programming case and will flag an error interrupt and abort the transfer. 0=Normal Length is used. 1=Length of transfer uses the DMA length rather than the standard control register length.	R/W	0h	RESET
2	Local DMA Tx Restart Address Enable When set, the DMA Channel's Start Address will reset to its initial value upon completion. This facilitates DMA Channel re-use without reprogramming. If this is not set, then the Start Address will be the last address accessed + transfer size upon completion. 0=On Completion: Start Address is last address accessed + transfer size. 1=On Completion: Start Address is reset to the initially programmed Start Address.	R/W	0h	RESET
1	Local DMA Tx Restart Enable This sets the DMA Channel to re-enable itself after a completion so the next DMA transfer can occur without requiring manual re-programming of the DMA Channel. 0=On Completion: DMA is disabled and needs to be restarted. 1=On Completion: DMA is re-enabled.	R/W	0h	RESET
0	Local DMA Tx Channel Enable This states that the DMA is programmed and ready to run. While this is cleared the QMSPI will be stalled, waiting for the DMA to being transferring, once the local FIFO is full. This is cleared by H/W once a transfer is completed. It can be re-set by H/W if Local DMA Restart is enabled. 0=The Local DMA Channel will not run. 1=The Local DMA Channel will run once the transfer requests this to function.	R/W	0h	RESET

30.12.42 QMSPI LOCAL DMA TX START ADDRESS CHANNEL 0 REGISTER

Offset	144h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Local DMA TX Start Address This enables the Local DMA usage (instead of the Central DMA) when the Descriptor Buffer register enables the DMA. Bit 0 is associated with Description Buffer[0] while bit 15 is associated with Description Buffer [15].	R/W	0h	RESET

© 2021 Microchip Technology Inc.

30.12.43 QMSPI LOCAL DMA TX LENGTH CHANNEL 0 REGISTER

Offset	148h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Local DMA Tx Length Address This is the maximum Length of the transfer in Bytes that the DMA Channel will allow access to. Once this length is reached the DMA Channel will terminate any further accesses, like the Central DMA does. This length can be used as a Byte Length to the QMSPI FSM's in the override mode.	R/W	0h	RESET

30.12.44 QMSPI LOCAL DMA TX CONTROL CHANNEL 1 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Tx Control Channel 0 Register.

30.12.45 QMSPI LOCAL DMA TX START ADDRESS CHANNEL 1 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Tx Start Address Channel 0 Register.

30.12.46 QMSPI LOCAL DMA TX LENGTH CHANNEL 1 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Tx Length Channel 0 Register.

30.12.47 QMSPI LOCAL DMA TX CONTROL CHANNEL 2 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Tx Control Channel 0 Register.

30.12.48 QMSPI LOCAL DMA TX START ADDRESS CHANNEL 2 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Tx Start Address Channel 0 Register.

30.12.49 QMSPI LOCAL DMA TX LENGTH CHANNEL 2 REGISTER

The format for this register is the same as the format of the QMSPI Local DMA Tx Length Channel 0 Register.

© 2021 Microchip Technology Inc.

31.0 INTERNAL MASTER SPI (IMSPI)

31.1 Overview

The EEC1727 includes a link interface intended for communication with a separate Thermal Monitor device. The link connects to the Thermal Monitor device so that registers in the external device appear in the internal address space of the EEC1727. The Internal Master SPI (IMSPI) link is not directly accessible by customers. It is configured at boot time by code in the Boot ROM.

31.2 References

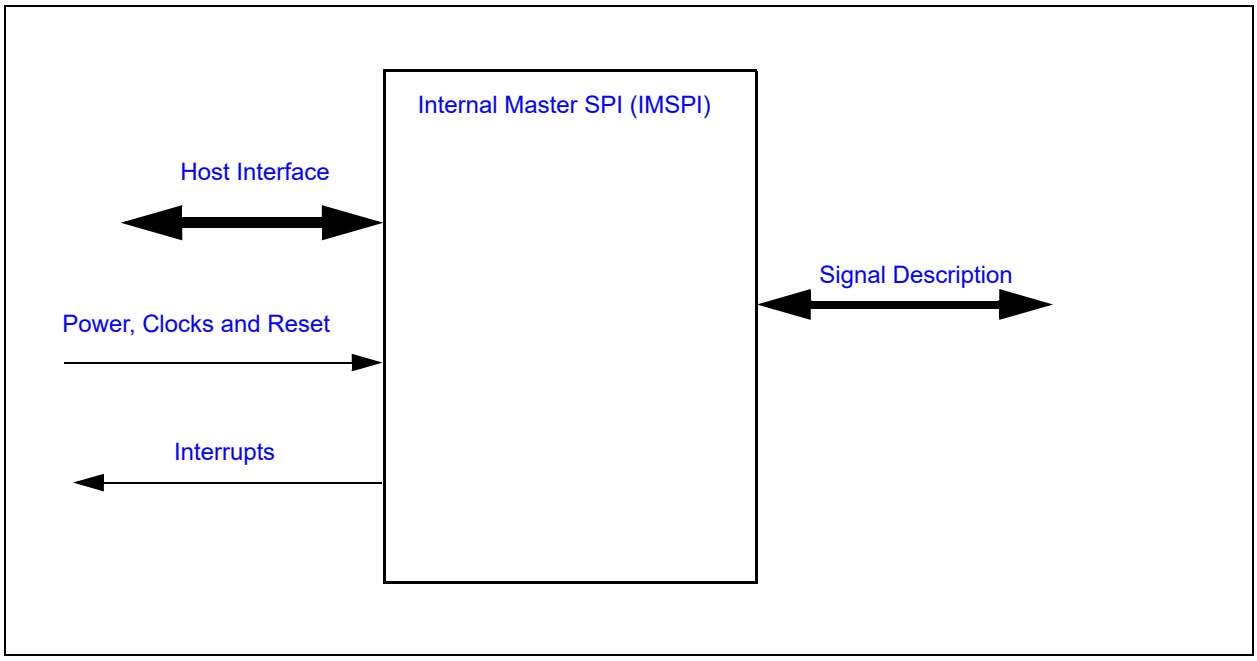
No references have been cited for this feature.

31.3 Terminology

There is no terminology defined for this section.

31.4 Interface

FIGURE 31-1: I/O DIAGRAM OF BLOCK



31.5 Signal Description

There are no external signals for this block.

31.6 Host Interface

The EEPROM interface is accessed by host software via a registered interface, as defined in Section 31.11, "EC Registers".

31.7 Power, Clocks and Reset

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

© 2021 Microchip Technology Inc.

31.7.1 POWER DOMAINS

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

31.7.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for IMSPI logic.

31.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.
RESET_ISPI	This reset is asserted either by the system reset or the soft reset bit in the controller register set.

31.8 Interrupts

Source	Description
IMSPI	IMSPI transfer terminated due to timeout. This interrupt is hidden from customers.

31.9 Low Power Modes

The ISPI Controller enters its lowest power state whenever it is not busy. If its sleep_enable input is asserted it will not start a new transfer.

31.10 Description

The IMSPI controller is a modified QMSPI controller used to map a block of address space into the internal address space of a companion device, using a modified SPI interface to communicate between the two devices.

31.11 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 Internal Master SPI (IMSPI) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 31-1: EC-ONLY REGISTER SUMMARY

EC Offset	Register Name
00h	IMSPI Mode Register
04h	IMSPI Status Register
08h	IMSPI Interrupt Enable Register
0Ch	IMSPI Timeout Control Register

31.11.1 IMSPI MODE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	R	-	-
25:24	IF_MODE This field sets the interface mode for the SPI controller. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single Mode	R/W	0h	RESET_ISPI
23:16	CLOCK_DIVIDE This SPI clock divide in terms of the number of system clocks. 255:1=The SPI clock period is equal to this number of system clocks 0=The SPI clock period is equal to 256 system clocks	R/W	0h	RESET_ISPI
15:11	Reserved	R	-	-
10	CPHA_MISO This field is the CPHA field of the underlying SPI controller which affects only the MISO Data. This field changes determines the clock edge on which data are captured, in combination with the CPOL field. For standard SPI Modes, this must be programmed with the same value as CPHA_MOSI. 1=If CPOL=0, data captured on Falling Edge; if CPOL=1, data captured on Rising Edge 0=If CPOL=0, data captured on Rising Edge; if CPOL=1, data captured on Falling Edge	R/W	0h	RESET_ISPI

© 2021 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9	CPHA_MOSI This field is the CPHA field of the underlying SPI controller which affects only the MOSI Data. This field changes determines the clock edge on which data are sent, in combination with the CPOL field. 1=If CPOL=0, data sent on Rising Edge; if CPOL=1, data sent on Falling Edge 0=If CPOL=0, data sent on Falling Edge; if CPOL=1, data sent on Rising Edge	R/W	0h	RESET_ISPI
8	CPOL This bit corresponds to the Polarity control for the underlying SPI controller. It describes the default state of the SPI Clock signal. 1=The clock starts in a high state 0=The clock starts in a low state	R/W	0h	RESET_ISPI
7:2	Reserved	R	-	-
1	SOFT_RESET A write of '1b' to this bit resets the controller. This bit is self-clearing.	W	-	RESET_ISPI
0	ACTIVATE This bit enables the controller. 1=The controller is enabled 0=The controller is disabled and placed in its lowest power state	R/W	0h	RESET_ISPI

31.11.2 IMSPI STATUS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	R	-	-

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	INVALID_RESPONSE The IMSPI has detected an invalid response field and therefore is aborting the transfer in failure. 1=A transfer error occurred due to an invalid response 0=No error occurred	R/WC	0h	RESET_ISPI
0	TIMEOUT This flags when a transfer has terminated due to timeout on the response phase. 1=A transfer error occurred due to an invalid response 0=No error occurred	R/WC	0h	RESET_ISPI

31.11.3 IMSPI INTERRUPT ENABLE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	R	-	-
1	INVALID_RESPONSE_IE Assert an EEPROM interrupt when the INVALID_RESPONSE status is asserted. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET_ISPI
0	TIMEOUT_IE Assert an IMSPI interrupt when the TIMEOUT status is asserted. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET_ISPI

© 2021 Microchip Technology Inc.

31.11.4 IMSPI TIMEOUT CONTROL REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	R	-	-
4:0	RESPONSE_TIMEOUT This field is the maximum number of response cycles the IMSPI will wait until flagging a timeout. A setting of 0 will disable the time-out feature.	R/W	0h	RESET_ISPI

32.0 TRACE FIFO DEBUG PORT (TFDP)

32.1 Introduction

The TFDP serially transmits Embedded Controller (EC)-originated diagnostic vectors to an external debug trace system.

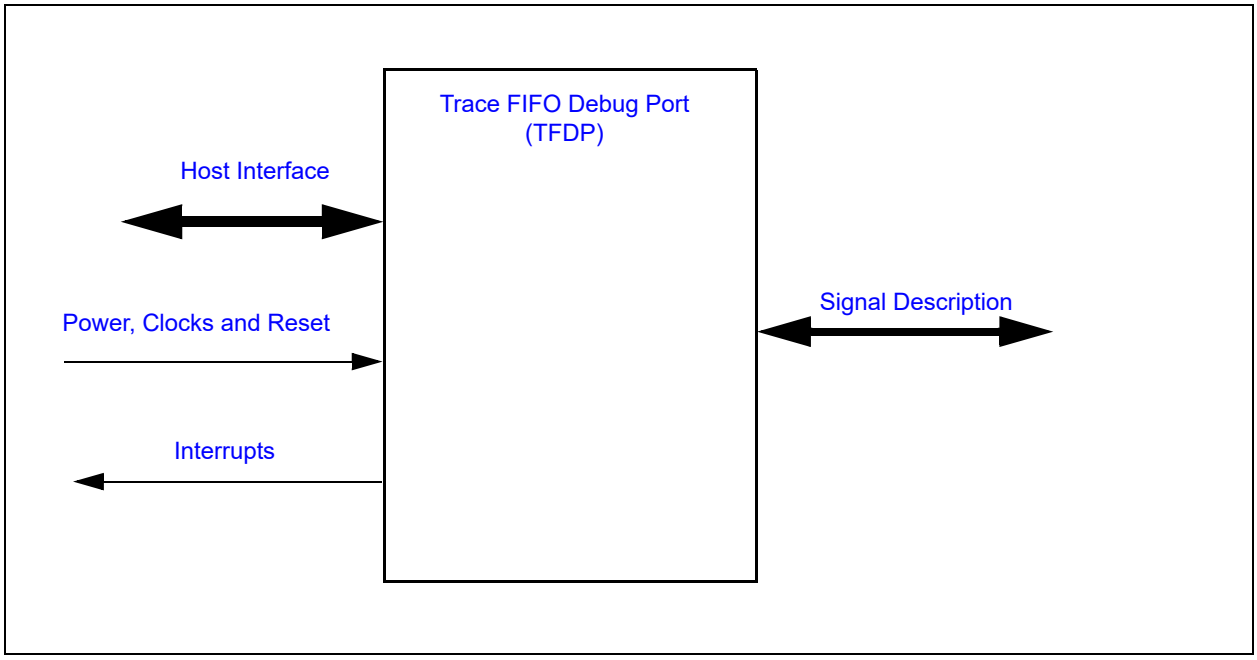
32.2 References

No references have been cited for this chapter.

32.3 Interface

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

FIGURE 32-1: I/O DIAGRAM OF BLOCK



32.4 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 32-1: SIGNAL DESCRIPTION

Name	Direction	Description
TFDP Clk	Output	Derived from EC Bus Clock.
TFDP Data	Output	Serialized data shifted out by TFDP Clk.

32.5 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".

© 2021 Microchip Technology Inc.

32.6 Power, Clocks and Reset

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

32.6.1 POWER DOMAINS

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

32.6.2 CLOCK INPUTS

Name	Description
48MHz	This is the main system clock.

32.6.3 RESETS

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

32.7 Interrupts

There are no interrupts generated from this block.

32.8 Low Power Modes

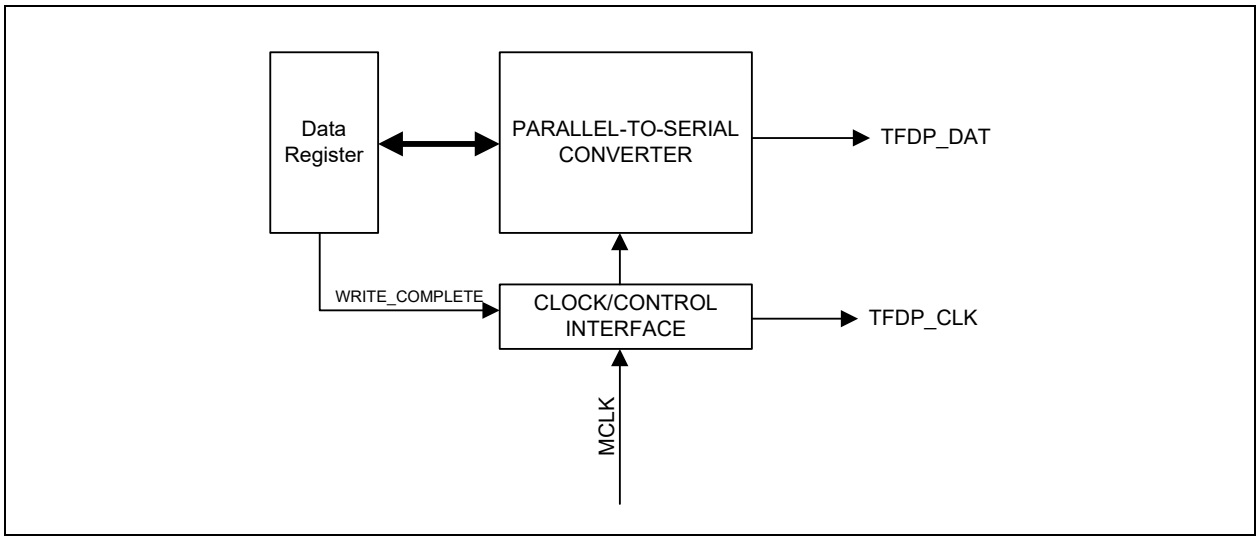
The Trace FIFO Debug Port (TFDP) may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

32.9 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 32-2.

FIGURE 32-2: BLOCK DIAGRAM OF TFDP DEBUG PORT

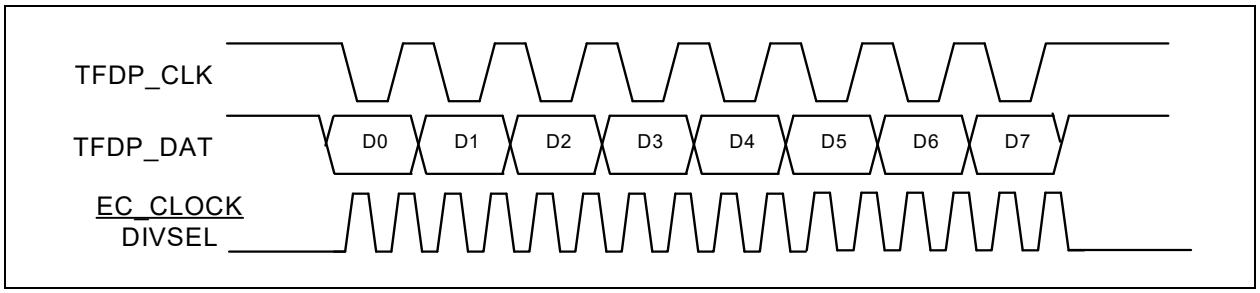


The firmware executing on the embedded controller writes to the Debug Data Register to initiate a transfer cycle (Figure 32-2). The Debug Data Register is loaded into a shift register and shifted out on TFDP_DAT LSB first at the programmed TFDP_CLK Clock rate (Figure 32-3).

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 32-3: DATA TRANSFER



32.10 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 32-2: REGISTER SUMMARY

Offset	Register Name
00h	Debug Data Register
04h	Debug Control Register

© 2021 Microchip Technology Inc.

32.10.1 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

32.10.2 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 32-3, "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 32-3: TFDP DEBUG CLOCKING

divsel	TFDP Debug Clock
00	24 MHz
01	12 MHz
10	6 MHz
11	Reserved

33.0 EC SUBSYSTEM REGISTERS

33.1 Introduction

This chapter defines a bank of registers associated with the EC Subsystem.

33.2 References

None

33.3 Interface

This block is designed to be accessed internally by the EC via the register interface.

33.4 Power, Clocks and Reset

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

33.4.1 POWER DOMAINS

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

33.4.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

33.4.3 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.

33.5 Interrupts

This block does not generate any interrupt events.

33.6 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.

33.7 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.

33.8 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".

© 2021 Microchip Technology Inc.

TABLE 33-1: 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	TEST
3Ch	TEST
40h	
44h	TEST
48h	TEST
50h	TEST
54h	Boot ROM Status Register
58h	TEST
5Ch	TEST
60h	TEST
64h	Reserved
68h	TEST
6Ch	TEST
70h	JTAG Master Configuration Register
74h	JTAG Master Status Register
78h	JTAG Master TDO Register
7Ch	JTAG Master TDI Register
80h	JTAG Master TMS Register
84h	JTAG Master Command Register
88h	TEST
90h	Virtual Wire Source Configuration Register
94h	Comparator Control Register
98h	Comparator Sleep Control Register
100h	TEST

© 2021 Microchip Technology Inc.

33.8.1 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

33.8.2 AHB ERROR CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
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

33.8.3 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

33.8.4 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

© 2021 Microchip Technology Inc.

33.8.5 DEBUG ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5	DEBUG_ENABLE_LOCK 1= ARM JTAG completely disabled. This means JTAG cannot be used for firmware/hardware debug. However, only Boundary Scan is accessible through JTAG port. 0= ARM JTAG accessible through JTAG.	R/W1S	0h	RESET_SYS
4	BOUNDARY SCAN PORT ENABLE 1= Enable Boundary scan port enable 0= Disable Boundary scan port enable If disabled, the Boundary scan Tap controller is not accessible via JTAG Port.	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 TRST# debug enable pin. 3=Reserved 2=The pins associated with the JTAG TCK and TMS switch to the debug interface when TRST# 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 TRST# 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 TRST# is de-asserted high. This setting should be used when the JTAG TAP controller is required for debugging	R/W	0h	RESET_SYS
0	DEBUG_EN This bit enables the JTAG/SWD debug port. 1=JTAG/SWD port enabled. A high on TRST# enables JTAG or SWD, as determined by SWD_EN 0=JTAG/SWD port disabled. JTAG/SWD cannot be enabled (the TRST# pin is ignored and the JTAG signals remain in their non-JTAG state)	R/W	0b	RESET_SYS
Note:	Boot ROM updates this register value on exit. Refer to the Boot ROM document for details.			

33.8.6 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.				

33.8.7 BOOT ROM STATUS REGISTER

Offset	54h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	–
1	WDT_EVENT WDT event status for Boot ROM	R/W1C	0	RESET_SYS-_nWDT
0	VTR_RESET_STATUS VTR_CORE reset status for Boot ROM	R/W1C	1	RESET_SYS

© 2021 Microchip Technology Inc.

33.8.8 JTAG MASTER CONFIGURATION REGISTER

Offset	70h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	–
3	MASTER_SLAVE This bit controls the direction of the JTAG port. 1=The JTAG Port is configured as a Master 0=The JTAG Port is configures as a Slave	R/W	0h	RESET_SYS
2:0	JTM_CLK This field determines the JTAG Master clock rate, derived from the 48MHz master clock. 7=375KHz 6=750KHz 5=1.5Mhz 4=3Mhz 3=6Mhz 2=12Mhz 1=24MHz 0=Reserved.	R/W	3h	RESET_SYS

33.8.9 JTAG MASTER STATUS REGISTER

Offset	74h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	R	-	–
0	JTM_DONE This bit is set to ‘1b’ when the JTAG Master Command Register is written. It becomes ‘0b’ when shifting has completed. Software can poll this bit to determine when a command has completed and it is therefore safe to remove the data in the JTAG Master TDO Register and load new data into the JTAG Master TMS Register and the JTAG Master TDI Register.	R	-	RESET_SYS

33.8.10 JTAG MASTER TDO REGISTER

Offset	78h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	JTM_TDO When the JTAG Master Command Register is written, from 1 to 32 bits are shifted into this register, starting with bit 0, from the JTAG_TDO pin. Shifting is at the rate determined by the JTM_CLK field in the JTAG Master Configuration Register	R/W	0h	RESET_SYS

33.8.11 JTAG MASTER TDI REGISTER

Offset	7Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	JTM_TDI When the JTAG Master Command Register is written, from 1 to 32 bits are shifted out of this register, starting with bit 0, onto the JTAG_TDI pin. Shifting is at the rate determined by the JTM_CLK field in the JTAG Master Configuration Register	R/W	0h	RESET_SYS

33.8.12 JTAG MASTER TMS REGISTER

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	JTM_TMS When the JTAG Master Command Register is written, from 1 to 32 bits are shifted out of this register, starting with bit 0, onto the JTAG_TMS pin. Shifting is at the rate determined by the JTM_CLK field in the JTAG Master Configuration Register	R/W	0h	RESET_SYS

© 2021 Microchip Technology Inc.

33.8.13 JTAG MASTER COMMAND REGISTER

Offset	84h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	R	-	–
4:0	JTM_COUNT If the JTAG Port is configured as a Master, writing this register starts clocking and shifting on the JTAG port. The JTAG Master port will shift JTM_COUNT+1 times, so writing a '0h' will shift 1 bit, and writing '31h' will shift 32 bits. The signal JTAG_CLK will cycle JTM_COUNT+1 times. The contents of the JTAG Master TMS Register and the JTAG Master TDI Register will be shifted out on the falling edge of JTAG_CLK and the.JTAG Master TDO Register will get shifted in on the rising edge of JTAG_CLK. If the JTAG Port is configured as a Slave, writing this register has no effect.	W	–	RESET_SYS

33.8.14 VIRTUAL WIRE SOURCE CONFIGURATION REGISTER

Offset	90h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2:0	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

33.8.15 COMPARATOR CONTROL REGISTER

Offset	94h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	Reserved	RES	-	-
4	Comparator 1 Enable 1= Enable Comparator 1 operation 0= Disable Comparator 1 operation	RW	0h	RESET_SYS
3	Reserved	RW	0h	RESET_SYS
2	Comparator 0 Configuration Locked 1= Configuration locked.Bits[2:0] are read only 0= Configuration not locked.Bits[2:0] are read write	R/W1X Note 2	CMP_STRAP0 pin = 1 then default= 1 All other configurations default= 0	RESET_SYS
1	Reserved	RES	0h	RESET_SYS
0	Comparator 0 Enable 1= Enable Comparator 0 operation 0= Disable Comparator 0 operation	RW or RO Note 1	CMP_STRAP0 pin = 1 then default= 1 All other configurations default= 0	RESET_SYS
Note 1: These bits become read only by writing bit 2 Comparator 0 Configuration Locked bit. **2:** If CMP_STRAP0 pin = 1, then Boot ROM writes this bit. Once this bit is written, this bit becomes read only.				

33.8.16 COMPARATOR SLEEP CONTROL REGISTER

Offset	98h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
1	Comparator 1 Deep Sleep Enable 0 = Comparator Deep Sleep Disable 1 = Comparator Deep Sleep Enable	R/W	0h	RESET_SYS
0	Comparator 0 Deep Sleep Enable 0 = Comparator Deep Sleep Disable 1 = Comparator Deep Sleep Enable	R/W or RO Note 1	0h	RESET_SYS
Note: Comparator Deep Sleep Enable must be set when the Comparator is enabled				

© 2021 Microchip Technology Inc.

34.0 SECURITY FEATURES

34.1 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

34.2 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

34.3 Terminology

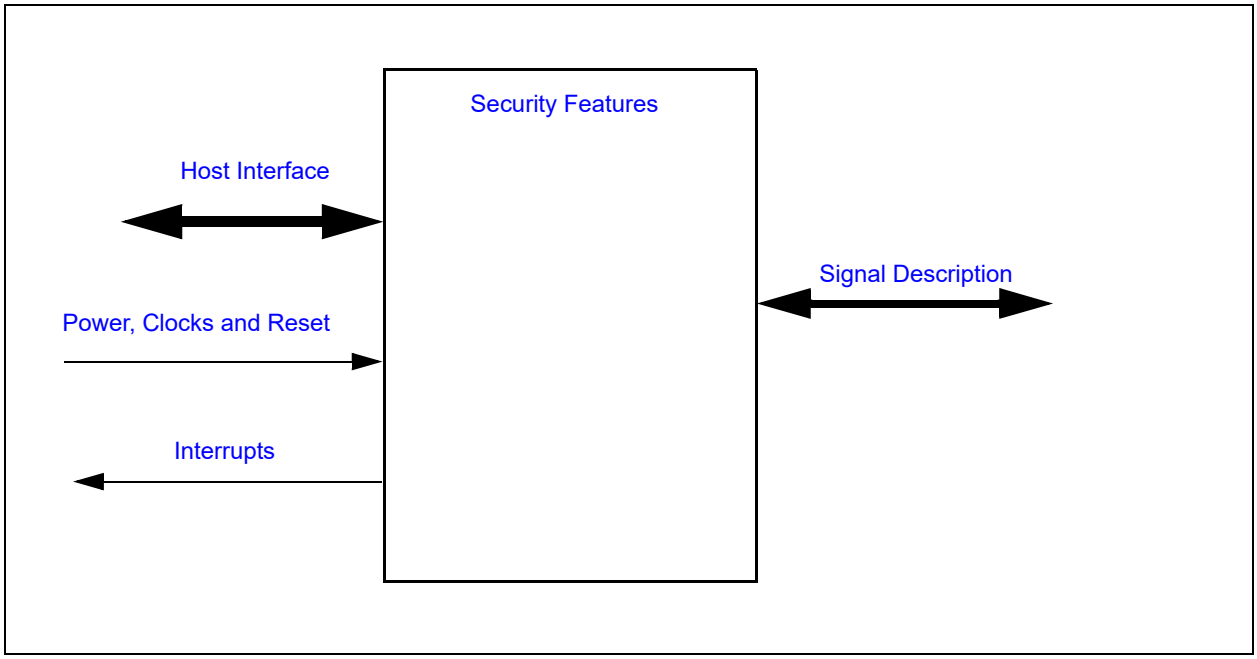
There is no terminology defined for this section.

34.4 Interface

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

© 2021 Microchip Technology Inc.

FIGURE 34-1: I/O DIAGRAM OF BLOCK



34.5 Signal Description

There are no external signals for this block.

34.6 Host Interface

Registers for the cryptographic hardware are accessible by the EC.

34.7 Power, Clocks and Reset

34.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The main power well used when the VBAT RAM is accessed by the EC.

34.7.2 CLOCK INPUTS

No special clocks are required for this block.

34.7.3 RESETS

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

© 2021 Microchip Technology Inc.

34.8 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

34.9 Low Power Modes

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

34.10 Description

The security hardware incorporates the following functions:

34.10.1 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), Cipher Feedback (CFB), Counter with CBC-MAC (CCM) and Galois/Counter Mode (GCM).

34.10.2 CRYPTOGRAPHIC HASHING

Standard SHA hash algorithms, including SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 are supported by hardware.

34.10.3 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, Curve448 and Edwards Curves
- 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
- ECC support for special curves Curve448 Ed25519 are inbuilt in hardware.
- Miller-Rabin primality testing

© 2021 Microchip Technology Inc.

The Public Key Engine includes a 8KB 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.

34.10.4 TRUE RANDOM NUMBER GENERATOR

A true Random Number Generator, which includes a 1K bit FIFO for pre-calculation of random bits. This block has Health Check function included with it.

34.10.5 MONOTONIC COUNTER

The Monotonic Counter is defined in Section 40.7.3, "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.

34.10.6 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.

34.11 Registers

There are no registers directly accessible to the application in this block. User must use the API's to use this block. Please refer to the Boot ROM document for the list of API's.

34.11.1 REGISTERS SUMMARY

The Public Key Engine, The Random Number Generator, the Hash Engine and the Symmetric Encryption Engine are all listed in the Block Overview and Base Addresses in Section 3.0, "Device Inventory".

© 2021 Microchip Technology Inc.

35.0 OTP BLOCK

35.1 Introduction

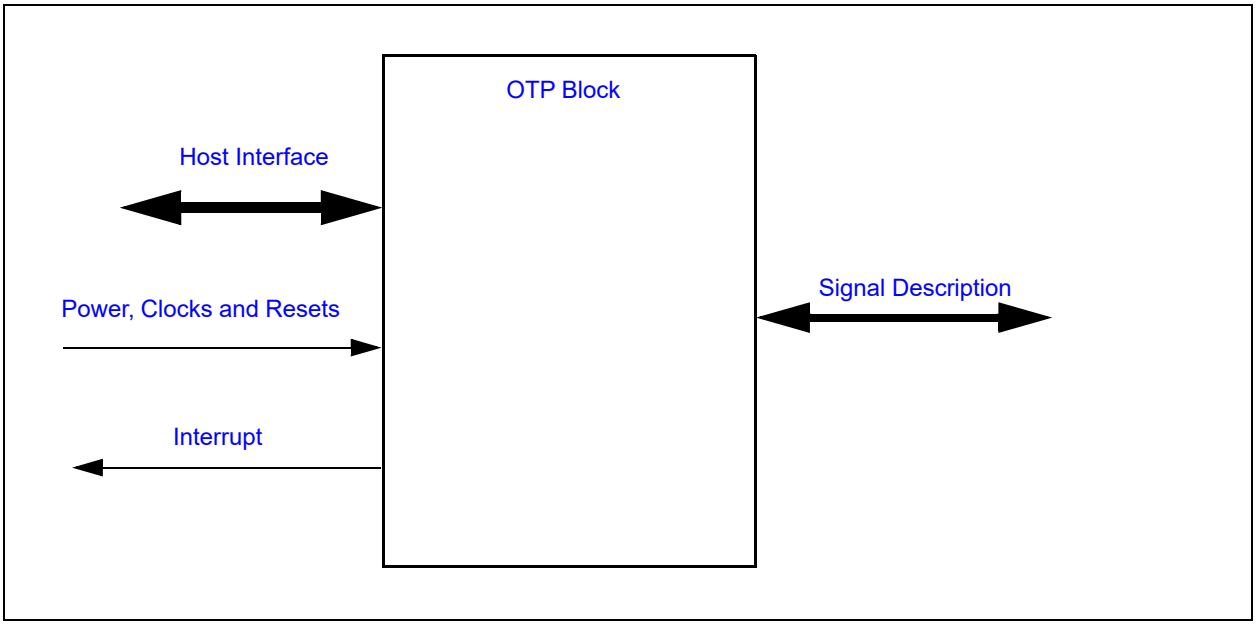
The OTP Block provides a means of programming and accessing a block of One Time Programmable memory.

35.2 Terminology

None.

35.3 Interface

FIGURE 35-1: OTP BLOCK INTERFACE DIAGRAM



35.4 Signal Description

There are no external signals from this block

35.5 Host Interface

The registers defined for the OTP Block are accessible by the EC.

35.6 Interrupt Interface

TABLE 35-1: INTERRUPT SIGNALS

Source	Description
OTP_READY	The OTP_READY interrupt will be generated whenever an OTP command is completed.

35.7 Power, Clocks and Resets

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

35.7.1 POWER DOMAINS

TABLE 35-2: 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.

35.7.2 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 35-3: CLOCKS

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

35.7.3 RESETS

TABLE 35-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.

35.8 Low Power Modes

The OTP always comes up in low power mode and stays in that state unless the firmware needs to use it

35.9 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.

35.10 OTP Memory Map

Please refer to Boot ROM document for this information.

TABLE 35-5: REGISTER SUMMARY

Offset	Register Name
44h	OTP Write Lock Register
48h	OTP Read Lock Register
4Ch	OTP Write Byte Lock Register
50h	OTP Read Byte Lock Register

© 2021 Microchip Technology Inc.

35.10.1 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

35.10.2 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

35.10.3 OTP WRITE BYTE LOCK REGISTER

Offset	4Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	OTP_WRITE_BYTE_LOCK Each bit locks write to a byte in the OTP range starting byte 320 to 351. 0=Not Locked 1=Locked	R/W1S	0h	RESET_SYS

35.10.4 OTP READ BYTE LOCK REGISTER

Offset	50h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	OTP_READ_BYTE_LOCK Each bit locks read to a byte in the OTP range starting byte 320 to 351. 0=Not Locked 1=Locked	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.

36.0 TEST MECHANISMS

36.1 JTAG Controller

The Controller, which is an IEEE compliant JTAG Port, has implemented all the mandatory JTAG instructions. This interface may be used to access the embedded controller's test access port (TAP).

36.1.1 INTERFACE

TABLE 36-1: JTAG PORT LIST

Signal Name	Direction	Description
JTAG_TCK	Input	Test Clock
JTAG_TMS	Input	Test Mode Select
JTAG_TDI	Input	Test Data In
JTAG_TDO	Output	Test Data Out (Note 36-1)
JTAG_RST#	Input	Test Reset, low active (Note 36-2)

Note 36-1 The JTAG_TDO output is the serial data output. It is presented on falling edges of TCK, 1/2 clock before each input shift, to provide setup and hold time to the next JTAG controller in the chain. The final TDO output pin, after all on-chip chaining is held in high-impedance mode (floating) except when valid data is being presented. The enabled/disabled state of the pin is also changed on falling edges of TCK.

Note 36-2 The JTAG_RST# input provides the Reset. Note that the reset state of the JTAG port is only local to the port: its effect is to keep the port in an idle state and to disengage it from the rest of the system, so that it does not affect other on-chip logic in this state.

TABLE 36-2: 2 PIN JTAG PORT LIST

Signal Name	Direction	Description
JTAG_TMS	Input	Test Mode Select
JTAG_TDO	Output	Test Data Out
JTAG_RST#	Input	Test Reset, low active

TABLE 36-3: SERIAL WIRE DEBUG PORT LIST

Signal Name	Direction	Description
Serial Wire Debug (SWD) See Debug Enable Register		
JTAG_TCK	Input	Test Clock
JTAG_TMS	Input	Test Mode Select
JTAG_RST#	Input	Test Reset, low active
Serial Wire Viewer (SWV) See Debug Enable Register		
JTAG_CLK	Input	Test Clock
JTAG_TMS	Input	Test Mode Select
JTAG_TDO	Output	Test Data Out
JTAG_RST#	Input	Test Reset, low active

36.1.2 POWER, CLOCKS, AND RESET

See power on sequence and reset timing.

36.1.2.1 Power Domains

TABLE 36-4: POWER SOURCES

Name	Description
VTR_CORE	The JTAG Controller logic and registers are implemented on this single power domain.

© 2021 Microchip Technology Inc.

36.1.2.2 Clocks

The JTAG port runs internally from the externally-provided JTAG_TCK clock pulses only. There is no requirement for JTAG_TCK to be constantly running.

36.1.2.3 Reset

The block has two resets: the JTAG_RST# input pin and Test-Logic-Reset as defined by the IEEE1149.1-19990 standard.

36.2 ARM Test Functions

TABLE 36-5: RESET SIGNALS

Name	Description
JTAG_RST#	The Test Reset Input from the pin interface used to reset all JTAG registers.

Test mechanisms for the ARM are described in Section 5.0, "ARM M4F Based Embedded Controller". if JTAG is enabled, hot plugging of JTAG connector is supported in the chip.

36.3 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 EEC1727 BSDL file. The EEC1727 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.

37.0 ELECTRICAL SPECIFICATIONS

37.1 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.

37.1.1 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

37.1.2 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
VTR3	1.8V Power Supply with respect to ground	-0.3V to +1.98V
VCC	3.3V Main Power Supply with respect to ground (Connected to VCC_PWRGD pin)	-0.3V to +3.63V

37.1.3 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

© 2021 Microchip Technology Inc.

37.2 Operational Specifications

37.2.1 POWER SUPPLY OPERATIONAL CHARACTERISTICS

TABLE 37-1: 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	3.3	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%.

37.2.2 AC ELECTRICAL SPECIFICATIONS

The AC Electrical Specifications for the clock input time are defined in Section 38.5, "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.

37.2.3 CAPACITIVE LOADING SPECIFICATIONS

The following table defines the maximum capacitive load validated for the buffer characteristics listed in Table 37-3, "DC Electrical Characteristics" and the AC characteristics defined in Section 38.5, "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 37-2: 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.

© 2021 Microchip Technology Inc.

37.2.4 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO-12 Type Buffer. See Note 3						
All PIO-12 Buffers Pull-up Resistor @3.3V @1.8V	R_{PU}	 34 35	 60 60	 95 105	 KΩ	Internal PU selected via the GPIO Pin Control Register.
All PIO-12 Buffers Pull-down Resistor @3.3V @1.8V	R_{PD}	 38 36	 63 63	 127 118	 KΩ	Internal PD selected via the GPIO Pin Control Register.
PIO-12 IOH at 1.8V for 10pf Load DRIVE_STRENGTH = 00b DRIVE_STRENGTH = 01b DRIVE_STRENGTH = 10b DRIVE_STRENGTH = 11b	 _ _ _ _	 2.02 4.03 8.06 12.1	 3.35 6.7 12.6 20	 5.26 10.5 21 31.5	 mA mA mA mA	The drive strength is determined by programming bits[5:4] of the Pin Control Register 2. Note 2 Note 2 Note 2 Note 2
PIO-12 IOL at 1.8V for 10pf Load DRIVE_STRENGTH = 00b DRIVE_STRENGTH = 01b DRIVE_STRENGTH = 10b DRIVE_STRENGTH = 11b	 _ _ _ _	 2.49 5.07 10.1 15.1	 4.5 9.16 18.2 27.3	 7.40 14.9 29.7 44	 mA mA mA mA	The drive strength is determined by programming bits[5:4] of the Pin Control Register 2. Note 2 Note 2 Note 2 Note 2
PIO-12 IOH at 3.3V for 10pf Load DRIVE_STRENGTH = 00b DRIVE_STRENGTH = 01b DRIVE_STRENGTH = 10b DRIVE_STRENGTH = 11b	 _ _ _ _	 4.04 8.01 16 24	 6 12 21 35.8	 8.58 17.1 34.2 51.3	 mA mA mA mA	The drive strength is determined by programming bits[5:4] of the Pin Control Register 2. Note 2 Note 2 Note 2 Note 2

© 2021 Microchip Technology Inc.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO-12 IOL at 3.3V for 10pf Load						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	4.77	7.2	10.1	mA	Note 2
DRIVE_STRENGTH = 01b	_	9.63	14.5	20.2	mA	Note 2
DRIVE_STRENGTH = 10b	_	19.2	26.4	40.3	mA	Note 2
DRIVE_STRENGTH = 11b	_	28.7	43.1	60	mA	Note 2
PIO-12 Rising Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	4.052	5.853	9.896	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	2.690	3.831	6.370	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	1.679	2.437	4.174	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	1.405	2.016	3.394	ns	For 1.8V at 10pf Load. See Note 2.
PIO-12 Falling Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	2.976	4.511	8.463	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	2.053	3.085	5.607	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	1.282	1.975	3.654	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	1.041	1.606	2.928	ns	For 1.8V at 10pf Load. See Note 2.
PIO-12 Rising Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	2.518	3.482	5.661	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	1.585	2.235	3.642	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	0.953	1.366	2.276	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	0.746	1.084	1.824	ns	For 3.3V at 10pf Load. See Note 2.

© 2021 Microchip Technology Inc.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO-12 Falling Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	2.017	2.809	4.833	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	1.220	1.754	3.082	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	0.679	1.008	1.837	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	0.498	0.715	1.404	ns	For 3.3V at 10pf Load. See Note 2.
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 (max)
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)
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 (max)
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)

© 2021 Microchip Technology Inc.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
IOD-4 mA Type Buffer	_				_	Same characteristics as an I and an OD-4mA.
O-8 mA Type Buffer Low Output Level High Output Level	 V_{OL} V_{OH}	 VTR - 0.4	 	 0.4 	 V V	 I_{OL} = 8 mA (max) 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 High Output Level	 V_{OL} V_{OH}	 VTR - 0.4	 	 0.4 	 V V	 I_{OL} = 12mA (max) 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.
PIO-24 Type Buffer. See Note 4						
All PIO-24 Buffers Pull-up Resistor @3.3V @1.8V	 R_{PU}	 34 35	 60 60	 95 105	 KΩ	Internal PU selected via the GPIO Pin Control Register.
All PIO-24 Buffers Pull-down Resistor @3.3V @1.8V	 R_{PD}	 38 36	 63 63	 127 118	 KΩ	Internal PD selected via the GPIO Pin Control Register.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO-24 IOH at 1.8V for 10pf Load						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	4.03	6.32	10.5	mA	Note 2
DRIVE_STRENGTH = 01b	_	8.05	12.6	20.9	mA	Note 2
DRIVE_STRENGTH = 10b	_	16.1	25.2	41.9	mA	Note 2
DRIVE_STRENGTH = 11b	_	24.1	37.8	62.6	mA	Note 2
PIO-24 IOL at 1.8V for 10pf Load						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	4.87	7.92	14.6	mA	Note 2
DRIVE_STRENGTH = 01b	_	10.1	18.3	29.7	mA	Note 2
DRIVE_STRENGTH = 10b	_	20	32.3	59	mA	Note 2
DRIVE_STRENGTH = 11b	_	30.1	54.3	88.4	mA	Note 2
PIO-24 IOH at 3.3V for 10pf Load						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	8.07	10.8	17.1	mA	Note 2
DRIVE_STRENGTH = 01b	_	16	23.8	34.2	mA	Note 2
DRIVE_STRENGTH = 10b	_	32	47.6	68.1	mA	Note 2
DRIVE_STRENGTH = 11b	_	47	71.1	101	mA	Note 2
PIO-24 IOL at 3.3V for 10pf Load						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	9.4	14.3	19.9	mA	Note 2
DRIVE_STRENGTH = 01b	_	19.2	28.8	40.2	mA	Note 2
DRIVE_STRENGTH = 10b	_	38.2	57.4	80	mA	Note 2
DRIVE_STRENGTH = 11b	_	57.2	85.9	119	mA	Note 2

© 2021 Microchip Technology Inc.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO-24 Rising Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	3.266	4.620	7.552	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	2.615	3.714	6.033	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	1.795	2.654	4.641	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	1.600	2.378	4.002	ns	For 1.8V at 10pf Load. See Note 2.
PIO-24 Falling Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	2.454	3.688	6.675	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	1.946	2.999	5.329	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	1.322	2.110	3.894	ns	For 1.8V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	1.103	1.796	3.258	ns	For 1.8V at 10pf Load. See Note 2.
PIO-24 Rising Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	1.781	2.590	4.288	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	1.273	1.872	3.189	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	0.855	1.256	2.180	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	0.711	1.048	1.822	ns	For 3.3V at 10pf Load. See Note 2.

© 2021 Microchip Technology Inc.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO-24 Falling Output Slope (pad)						The drive strength is determined by programming bits[5:4] of the Pin Control Register 2.
DRIVE_STRENGTH = 00b	_	1.373	2.023	3.617	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 01b	_	0.884	1.339	2.552	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 10b	_	0.538	0.821	1.618	ns	For 3.3V at 10pf Load. See Note 2.
DRIVE_STRENGTH = 11b	_	0.417	0.641	1.262	ns	For 3.3V at 10pf Load. See Note 2.
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-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA (max)
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 (max)
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)

© 2021 Microchip Technology Inc.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
IOD-8 mA Type Buffer	_				_	Same characteristics as an I and an OD-8mA.
O-16 mA Type Buffer Low Output Level High Output Level	 V_{OL} V_{OH}	 VTR - 0.4	 	 0.4 	 V V	 I_{OL} = 16 mA (max) I_{OH} = -16 mA (min)
IO-16 mA Type Buffer	_				_	Same characteristics as an I and an O-16mA.
OD-16 mA Type Buffer Low Output Level	 V_{OL}	 	 	 0.4	 V	 I_{OL} = 16 mA (min)
IOD-16 mA Type Buffer	_				_	Same characteristics as an I and an OD-16mA.
O-24 mA Type Buffer Low Output Level High Output Level	 V_{OL} V_{OH}	 VTR - 0.4	 	 0.4 	 V V	 I_{OL} = 24mA (max) I_{OH} = -24mA (min)
IO-24 mA Type Buffer	_				_	Same characteristics as an I and an O-24mA.
OD-24 mA Type Buffer Low Output Level	 V_{OL}	 	 	 0.4	 V	 I_{OL} = 24mA (min)
IOD-24 mA Type Buffer	_				_	Same characteristics as an I and an OD-24mA.
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.

TABLE 37-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
ADC Reference Pins						
ADC_VREF						
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 1: Tolerance for the pins are not 5VT Unless the pin chapter explicitly indicates specific pin has “Over-voltage protection” feature.

2: These values are guaranteed by design and not tested in production test.

3: In the Table 2-2, "EEC1727 68 WFBGA PIN MUX TABLE" PIO-12 buffer type are represented as PIO with empty drive strength column.

4: In the Table 2-2, "EEC1727 68 WFBGA PIN MUX TABLE" PIO-24 buffer type are represented as PIO with 24mA in the drive strength column.

37.2.4.1 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 37-4: PIN LEAKAGE (VTR=3.3V + 5%; VTR = 1.8V +5%)

(T_A = -40°C to +85°C)						
Leakage Current	I_{IL}			+/-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:

© 2021 Microchip Technology Inc.

TABLE 37-5: 5V TOLERANT LEAKAGE CURRENTS (VTR = 3.3V-5%)

(T_A = -40°C to +85°C)						
Three-State Input Leakage Current for 5V Tolerant Pins	I_{IL}	-	-	+/-2	µA	VIN = 0 to 5.5V

Note: These measurements are done without an external pull-up.

TABLE 37-6: 3.6V TOLERANT LEAKAGE CURRENTS (VTR = 1.8V-5%)

(T_A = -40°C to +85°C)						
Three-State Input Leakage Current for Under-Voltage Tolerant Pins	I_{IL}	-	-	+/-2	µA	VIN=0 to 3.6V

Note: This measurements are done without an external pull-up.

BACKDRIVE PROTECTION

TABLE 37-7: BACKDRIVE PROTECTION LEAKAGE CURRENTS (VTR=0V)

(T_A = -40°C to +85°C)						
Input Leakage	I_{IL}			+/-3	µA	0V < VIN ≤ 5.5V

37.2.5 ADC ELECTRICAL CHARACTERISTICS

TABLE 37-8: 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
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	

TABLE 37-8: ADC CHARACTERISTICS (CONTINUED)

Symbol	Parameter	MIN	TYP	MAX	Units	Comments
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	

37.2.6 THERMAL CHARACTERISTICS

TABLE 37-9: THERMAL OPERATING CONDITIONS

Rating	Symbol	MIN	TYP	MAX	Unit
Consumer Temperature Devices					
Operating Junction Temperature Range	T_J		—	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

TABLE 37-10: THERMAL PACKAGING CHARACTERISTICS

Characteristics	Symbol	TYP	MAX	Unit	Part #
Package Thermal Resistance, 68-pin WFBGA	θ_{JA}		—	°C/W	**EEC1727**
	θjC		—	°C/W	
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.					

© 2021 Microchip Technology Inc.

37.3 Power Consumption

TABLE 37-11: VBAT SUPPLY CURRENT, I_VBAT (VBAT=3.3V)

VCC	VTR	96 MHz	Typical (3.3V, 25^0 C)	Max (3.3V, 25^0 C)	Units	Comments
Off	Off	Off	8.0	20.0	mA	Internal 32kHz oscillator - supplied by coin cell
Off	On	Off	5.0	6.0	mA	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.

38.0 TIMING DIAGRAMS

Note: Timing values are preliminary and may change after characterization.

38.1 Power-up and Power-down Timing

FIGURE 38-1: VTR/VBAT POWER-UP TIMING

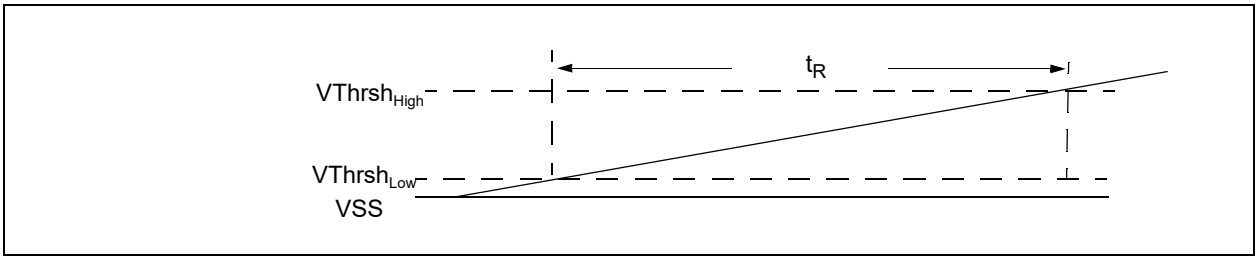


FIGURE 38-2: VTR RESET AND POWER-DOWN

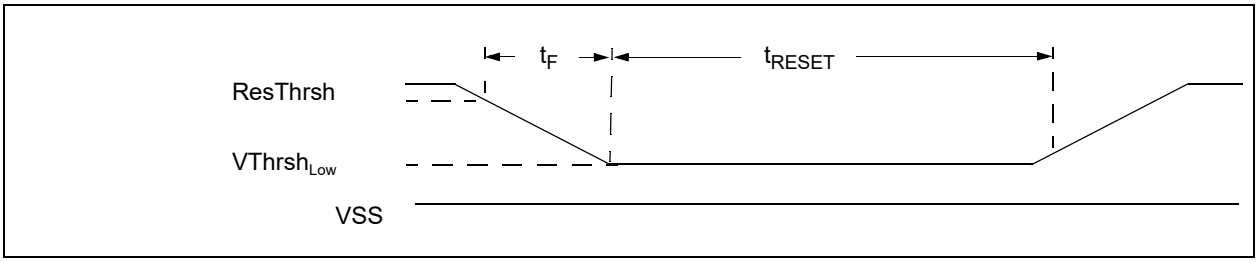


TABLE 38-1: VTR/VBAT TIMING PARAMETERS

Symbol	Parameter	MIN	TYP	MAX	Units	Notes
t_F	VTR Fall time	30			µs	1
	VBAT Fall time	30			µs	
t_R	VTR Rise time	0.050		20	ms	1
	VBAT Rise time	0.100		20	ms	
t_{RESET}	Minimum Reset Time	1			µs	
$VThrsh_{Low}$	VTR Low Voltage Threshold	0.3			V	1
	VBAT Low Voltage Threshold	0.3			V	
$VThrsh_{High}$	VTR High Voltage Threshold			2.5	V	1
	VBAT High Voltage Threshold			2.5	V	
ResThrsh	VTR Reset Threshold	0.5	1.8	2.7	V	1
	VBAT Reset Threshold	0.4	1.25	1.9	V	

Note 1: VTR applies to both VTR_REG and VTR_ANALOG

© 2021 Microchip Technology Inc.

38.2 Power Sequencing

FIGURE 38-3: POWER RAIL SEQUENCING

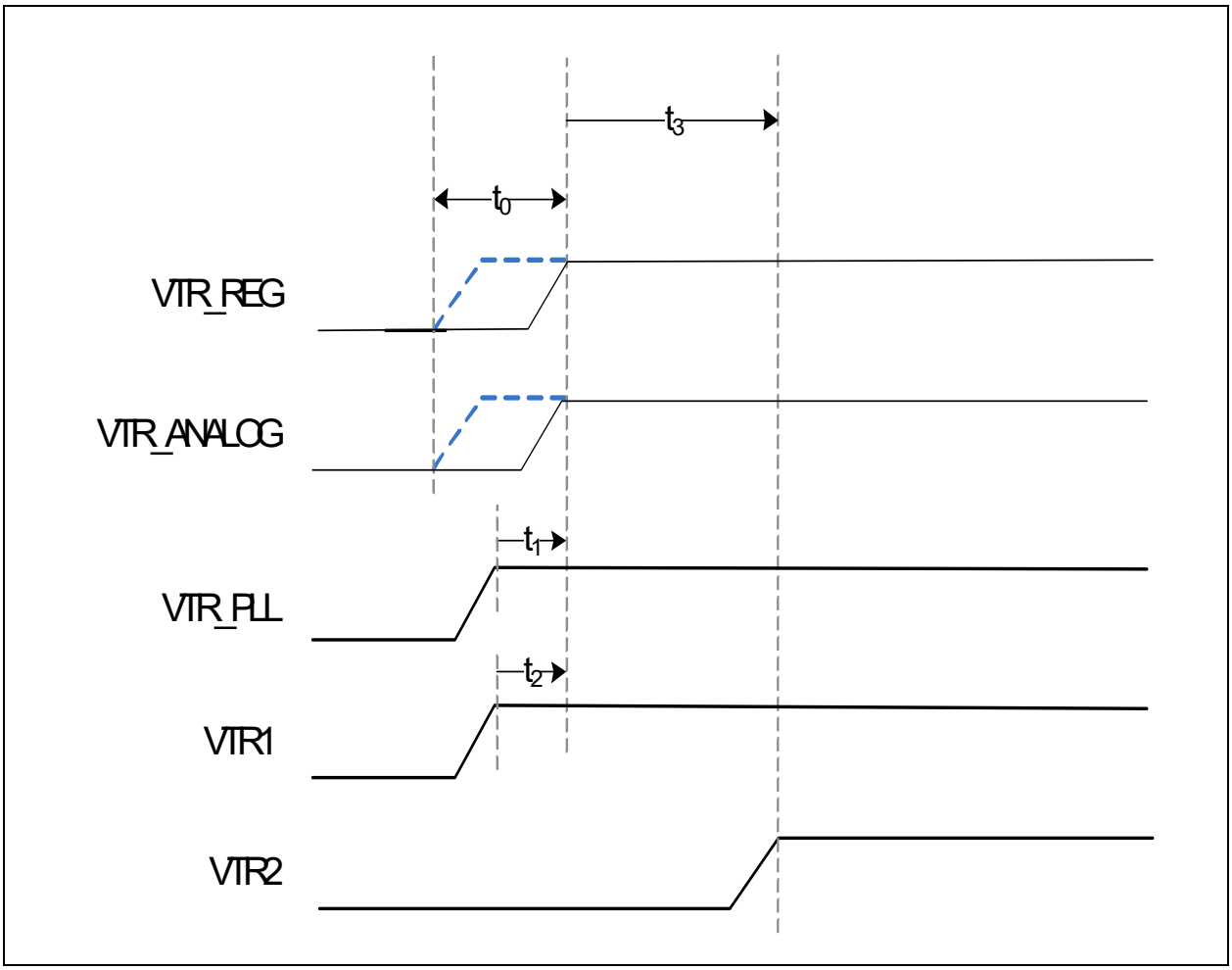


TABLE 38-2: POWER SEQUENCINGPARAMETERS

Symbol	Parameter	Min	Typ	Max	Units	Notes
t_0	VTR_ANALOG above minimum operating threshold to VTR_REG above minimum operating threshold	0		1	ms	1, 3
	VTR_REG above minimum operating threshold to VTR_ANALOG above minimum operating threshold	0		1	ms	
t_1	VTR_PLL above minimum operating threshold to VTR_ANALOG above minimum operating threshold			0	ms	2, 3

TABLE 38-2: POWER SEQUENCINGPARAMETERS (CONTINUED)

Symbol	Parameter	Min	Typ	Max	Units	Notes
t_2	VTR_ANALOG and VTR_REG are both above minimum operating thresholds to VTR1 above minimum operating threshold.	0		1	ms	2, 3
t_3	VTR_ANALOG and VTR_REG are both above minimum operating thresholds to VTR2 above minimum operating threshold. VTR2 at1.8V(nom) or 3.3V(nom)	0		1	ms	2, 3

Note 1: VTR_ANALOG and VTR_REG may ramp in either order

2: The SHD_CS# pin, which is powered by VTR2, must be powered before the Boot ROM samples this pin.

3: Minimum operating threshold values for Power Rails are defined in Table 37-1, "Power Supply Operating Conditions".

Please refer Boot ROM documentation for complete power sequencing options and timing requirements.

© 2021 Microchip Technology Inc.

38.3 Boot from SPI Flash Timing

Refer to EEC1727 Boot ROM document for the sequence and timing.

38.4 nRESET_IN Timing

FIGURE 38-4: NRESET_IN TIMING

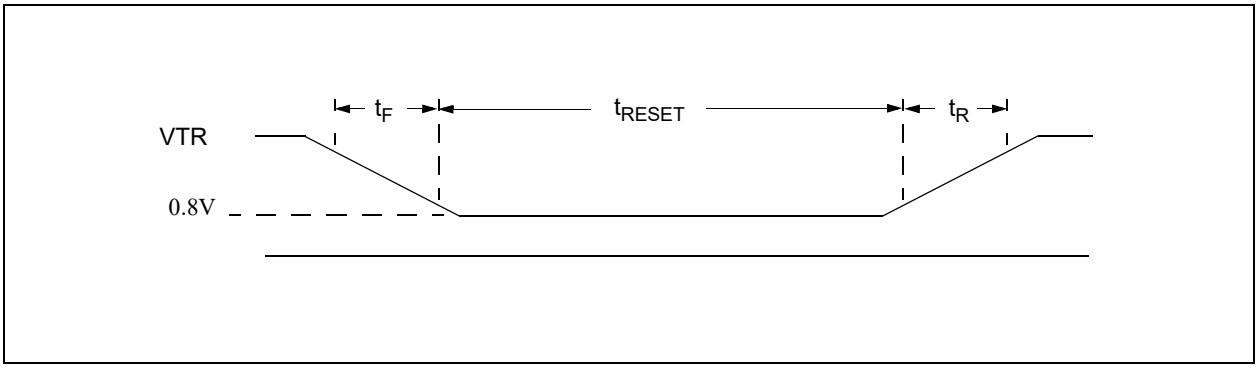


TABLE 38-3: RESETI# TIMING PARAMETERS

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.					

© 2021 Microchip Technology Inc.

38.5 Clocking AC Timing Characteristics

FIGURE 38-5: CLOCK TIMING DIAGRAM

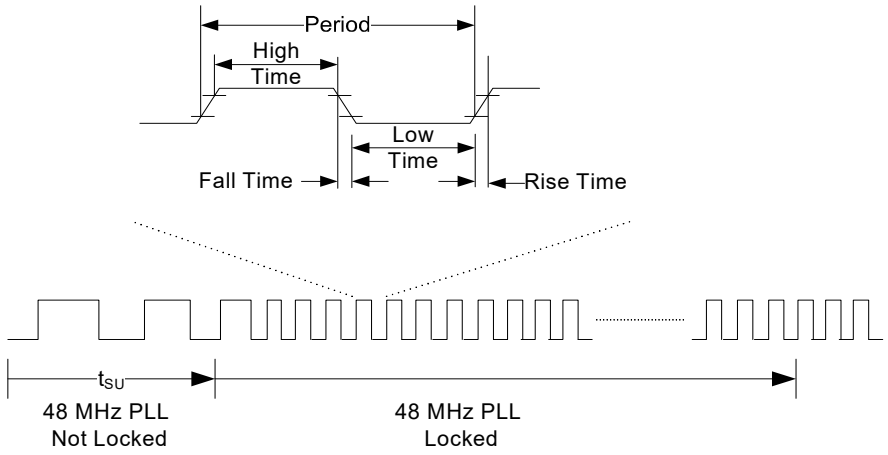


TABLE 38-4: CLOCK TIMING PARAMETERS

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 6)	-	-	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	16	-	48	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: See the PCB Layout guide for design requirements and recommended 32.768 kHz Crystal Oscillators.

4: An external single-ended 32KHz clock is required to have an accuracy of +/- 100 ppm.

5: The external single-ended 32KHz clock source may be connected to either the SUSCLK_IN pin or 32KHZ_IN pin.

6: PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset.

TABLE 38-4: CLOCK TIMING PARAMETERS (CONTINUED)

Clock	Symbol	Parameters	MIN	TYP	MAX	Units
32.768 kHz Crystal Oscillator (Note 3)	-	Operating Frequency	-	32.768	-	kHz
32KHz Silicon Oscillator	-	Operating Frequency	32.112	32.768	33.424	kHz
	-	Start-up delay from 0k Hz to Operating Frequency			150	us
32KHz single-ended input (Note 5)	-	Operating Frequency	-	32.768	-	kHz
	-	Period	(Note 4)	30.52	(Note 4)	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: See the PCB Layout guide for design requirements and recommended 32.768 kHz Crystal Oscillators.

4: An external single-ended 32KHz clock is required to have an accuracy of +/- 100 ppm.

5: The external single-ended 32KHz clock source may be connected to either the SUSCLK_IN pin or 32KHZ_IN pin.

6: PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset.

© 2021 Microchip Technology Inc.

38.6 GPIO Timings

FIGURE 38-6: GPIO TIMING

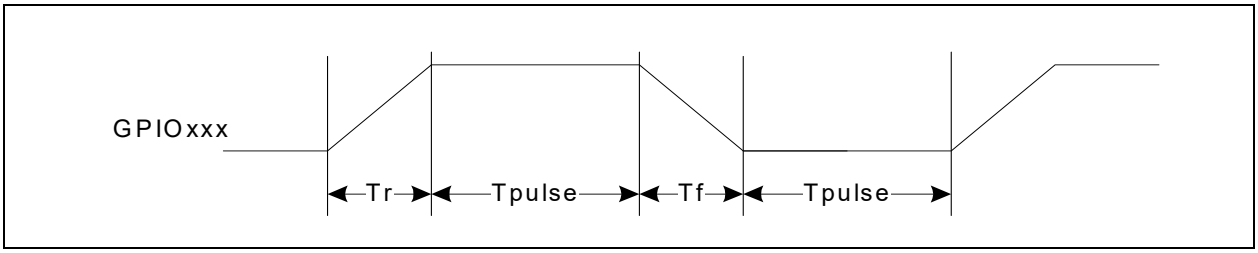


TABLE 38-5: GPIO TIMING PARAMETERS

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

© 2021 Microchip Technology Inc.

38.7 Serial Port (UART) Data Timing

FIGURE 38-7: SERIAL PORT DATA

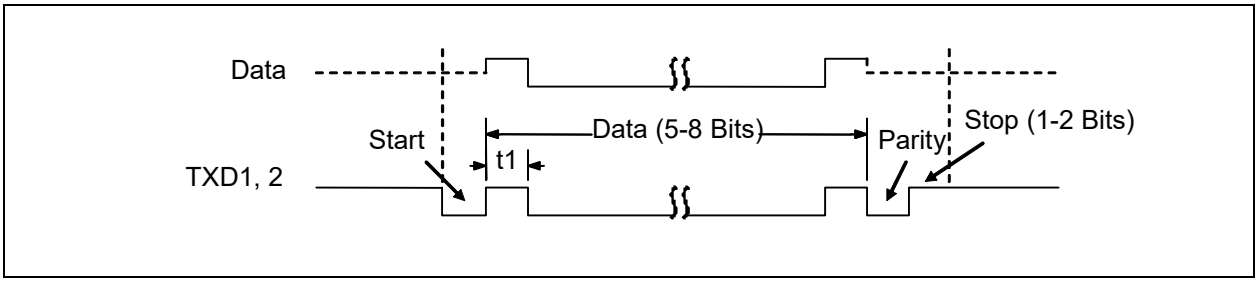


TABLE 38-6: SERIAL PORT DATA PARAMETERS

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 13-8, "UART Baud Rates using Clock Source 1.8432MHz" and Table 13-9, "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.

© 2021 Microchip Technology Inc.

38.8 PWM Timing

FIGURE 38-8: PWM OUTPUT TIMING

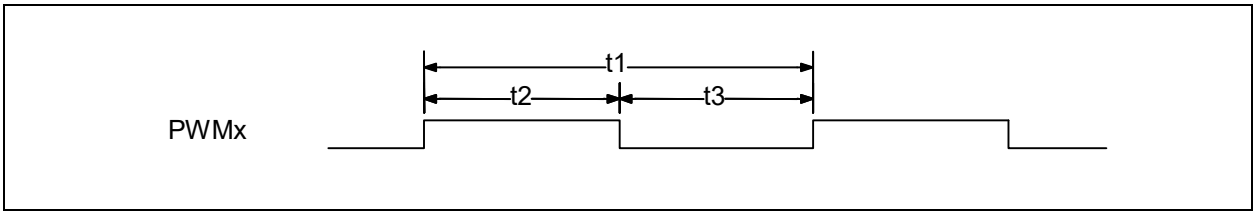


TABLE 38-7: PWM TIMING PARAMETERS

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	%

38.9 Fan Tachometer Timing

FIGURE 38-9: FAN TACHOMETER INPUT TIMING

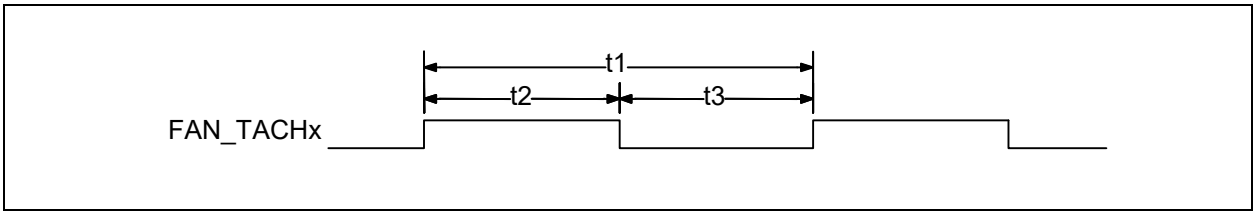


TABLE 38-8: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time				µsec
t2	Pulse High Time				
t3	Pulse Low Time				
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.				

© 2021 Microchip Technology Inc.

38.10 Blinking/Breathing PWM Timing

FIGURE 38-10: BLINKING/BREATHING PWM OUTPUT TIMING

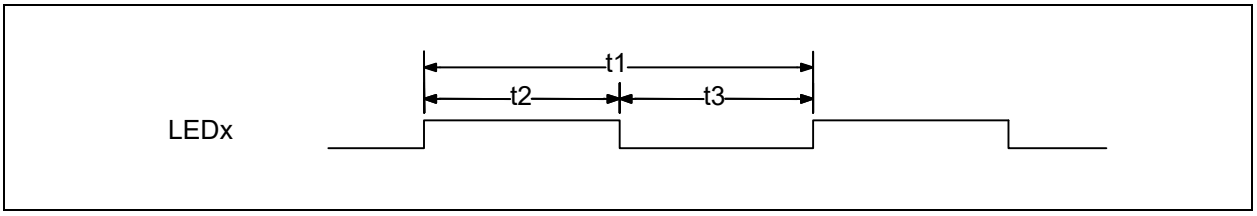


TABLE 38-9: BLINKING/BREATHING PWM TIMING PARAMETERS, BLINKING MODE

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 38-10: 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	%

38.11 I2C/SMBus Timing

FIGURE 38-11: I2C/SMBUS TIMING

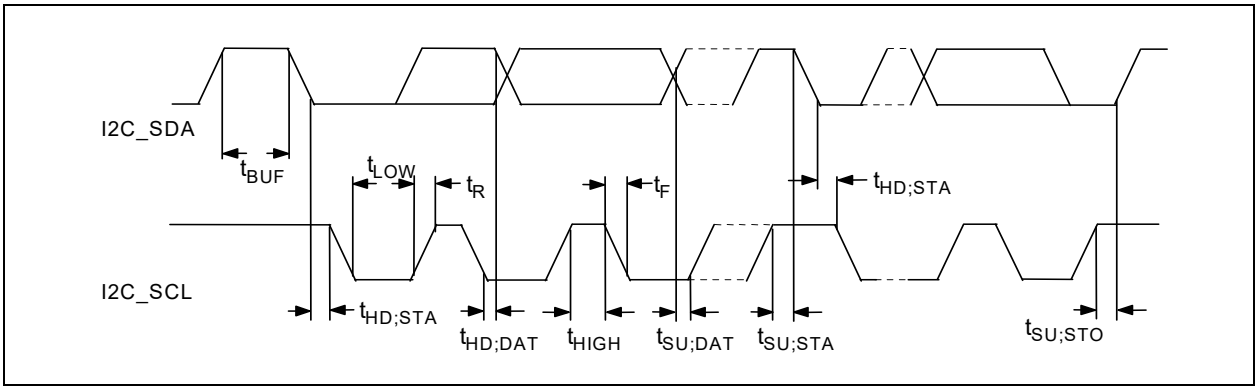


TABLE 38-11: I2C/SMBUS TIMING PARAMETERS

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

© 2021 Microchip Technology Inc.

38.12 Quad SPI Master Controller - Serial Peripheral Interface (QMSPI) Timings

FIGURE 38-12: SPI CLOCK TIMING

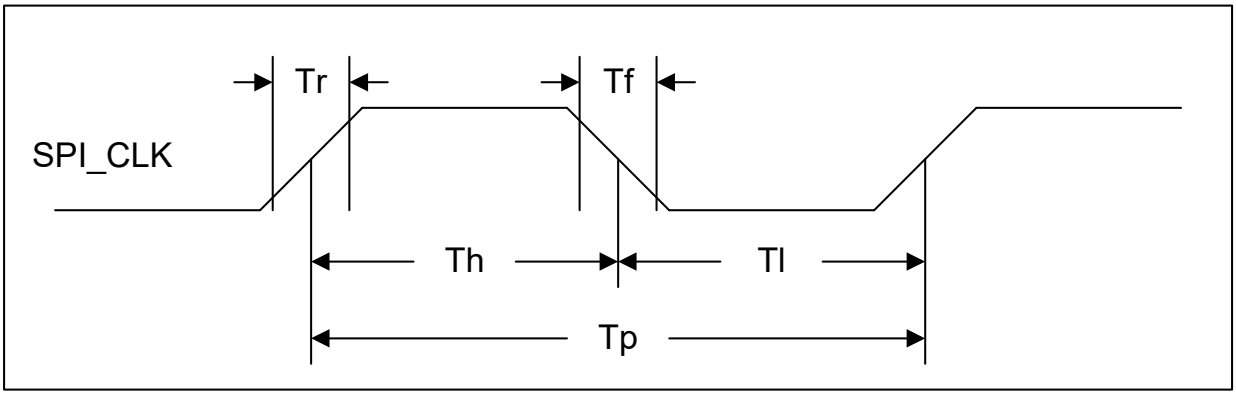
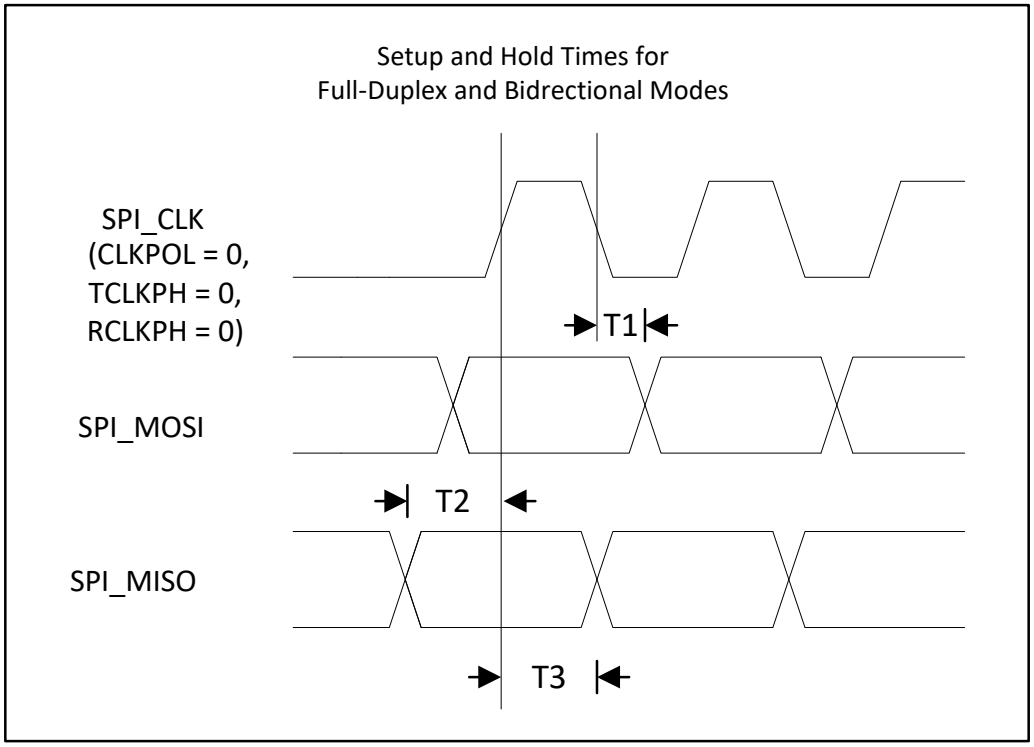


TABLE 38-12: SPI CLOCK TIMING PARAMETERS

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:	Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow.				

FIGURE 38-13: SPI SETUP AND HOLD TIMES



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 38-13: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			2	ns
T2	Data IN Setup Time	5.5			ns
T3	Data IN Hold Time	0			ns
Note:	Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow.				

© 2021 Microchip Technology Inc.

38.13 Serial Debug Port Timing

FIGURE 38-14: SERIAL DEBUG PORT TIMING PARAMETERS

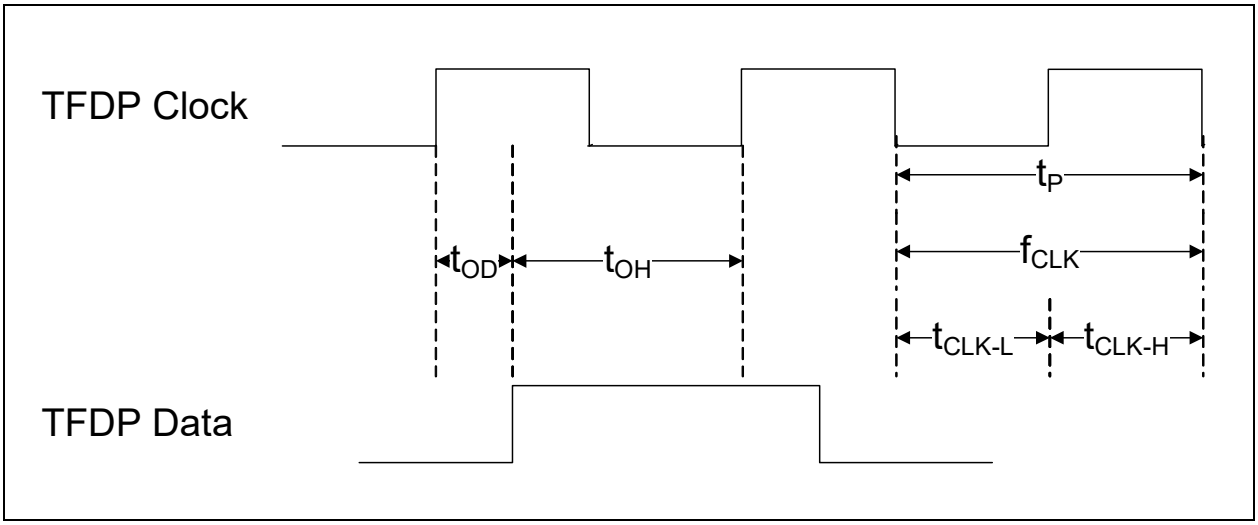


TABLE 38-14: SERIAL DEBUG PORT INTERFACE TIMING PARAMETERS

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 $t_{CLK-L} = t_{CLK-H}$.

38.14 JTAG Interface Timing

FIGURE 38-15: JTAG POWER-UP & ASYNCHRONOUS RESET TIMING

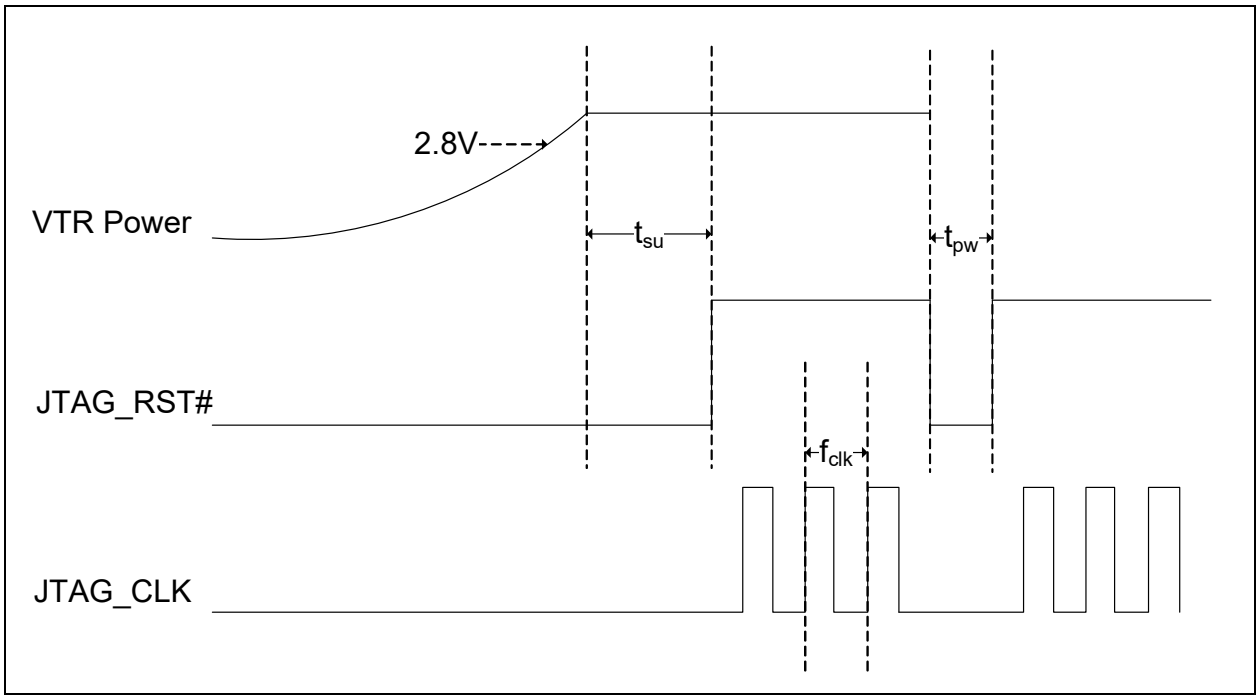
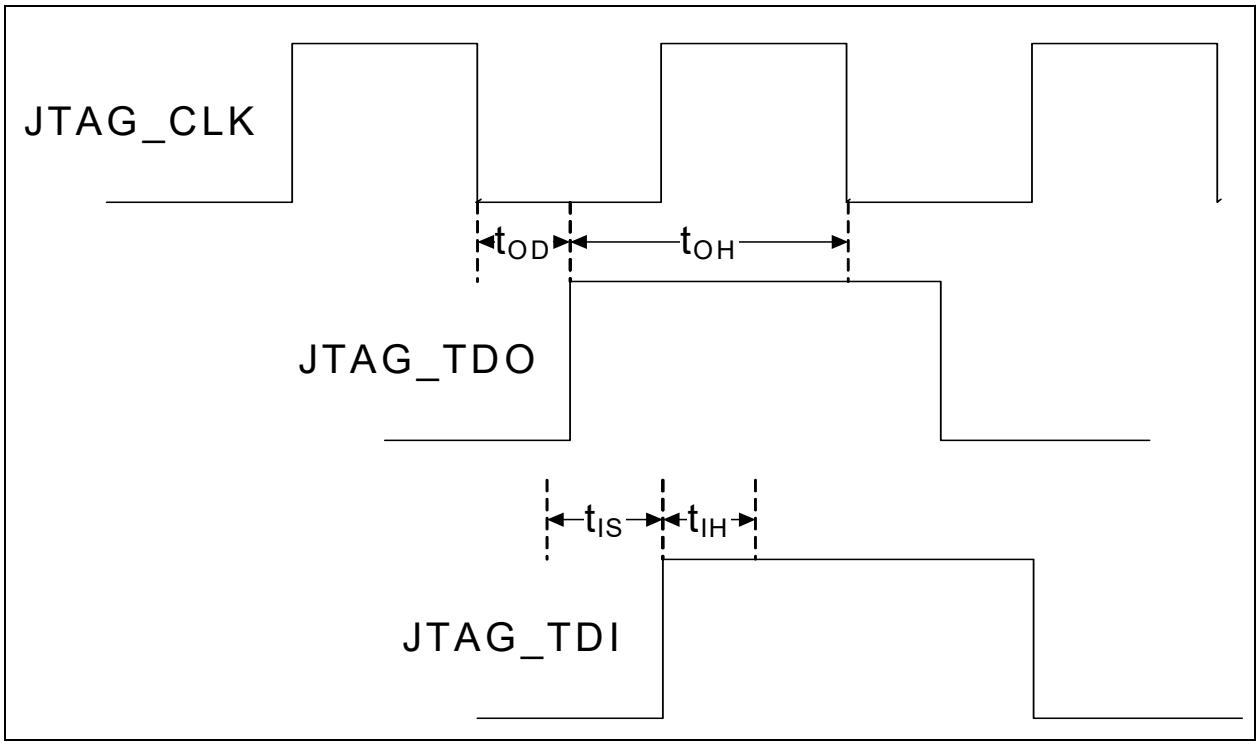


FIGURE 38-16: JTAG SETUP & HOLD PARAMETERS



© 2021 Microchip Technology Inc.

TABLE 38-15: 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

Revision	Section/Figure/Entry	Correction
DS00003840A (02-08-21)	Preliminary Document	

© 2021 Microchip Technology Inc.

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

© 2021 Microchip Technology Inc.

PRODUCT IDENTIFICATION SYSTEM

Not all of the possible combinations of Device, Temperature Range and Package may be offered for sale. 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	-	X/XXX[2]	-	[X][3]
Device		Total SRAM		Version/ Revision		Temp Range/ Package		Tape and Reel Option

Device:	EEC1727[1]		Embedded Controller with Authentication enabled
Total SRAM	Not Used		
Version/ Revision:	Not Used		
Temperature Range	I/	=	-40°C to +85°C (Industrial)
Package:	2GW		68 pin WFBGA[2], 6mm x 6mm body, 0.65mm pitch
Tape and Reel Option:	Blank TR	= =	Tray packaging Tape and Reel[3]

Example:

a) EEC1727-I/2GW = EEC1727, 416KB total SRAM, Customer "A" ROM Standard ROM, ROM Version 1, 68- WFBGA 6x6mm body, Industrial grade, 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

© 2021 Microchip Technology Inc.

Note the following details of the code protection feature on Microchip devices:

- 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 and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- 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. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like 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.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES 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 INDIRECT, 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, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, 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, 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, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, 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, 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, and Symmcom 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.

© 2021, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 9781522476283

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

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

© 2021 Microchip Technology Inc.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

EEC1727-I/2GW EEC1727-I/2GW-TR