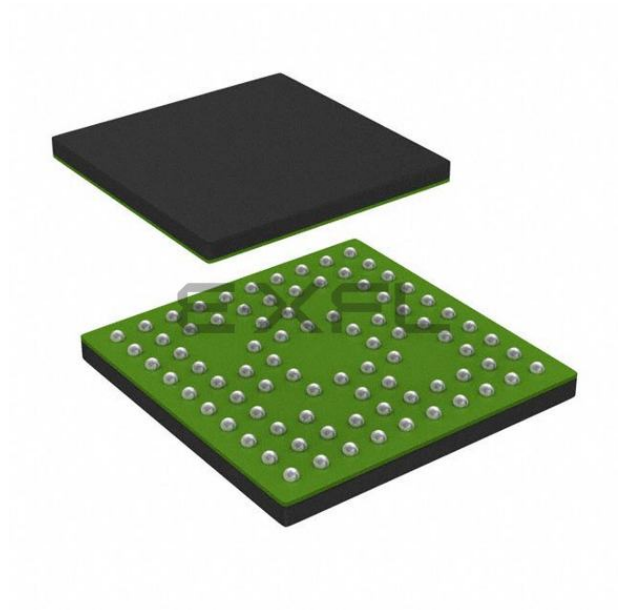


Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Active
Applications	Cryptography
Core Processor	ARM® Cortex®-M4F
Program Memory Type	-
Controller Series	-
RAM Size	480KB
Interface	I²C, SPI, UART
Number of I/O	65
Voltage - Supply	1.71V ~ 3.465V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	84-WFBGA
Supplier Device Package	84-WFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/cec1702q-b1-sx-tr

CEC1702-84	Pin Name	Power Rail	Pad Type	Glitch Prot	Over-voltage Prot	Under-voltage Prot	Backdrive Prot
K3	GPIO224/QSPI0_IO1	VTR2	PIO	X		X	X
K4	GPIO227/QSPI0_IO2	VTR2	PIO	X		X	X
K5	GPIO223/QSPI0_IO0	VTR2	PIO	X		X	X
K6	GPIO056/PWM3/QSPI0_CLK	VTR2	PIO	X		X	X
K7	GPIO055/PWM2/QSPI0_CS#	VTR2	PIO	X		X	X
K8	GPIO040/KSO00	VTR2	PIO	X		X	X
K9	GPIO026/TIN1/KSI3	VTR2	PIO	X		X	X
K10	GPIO053/PWM0/GPWM0	VTR2	PIO	X	X	X	X

2.6 Signal Description by Signal

EMULATED POWER WELL

Power well emulation for GPIOs and for signals that are multiplexed with GPIO signals is controlled by the POWER_-GATING field in the GPIO Pin Control Register. Power well emulation for signals that are not multiplexed with GPIO signals is defined by the entries in this column.

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.

Signal	Emulated Power Rail	Gated State	Notes
ADC00	VTR	Low	
ADC01	VTR	Low	
ADC02	VTR	Low	
ADC03	VTR	Low	
ADC04	VTR	Low	
BGND		Low	
BGPO0	VTR	Low	
CTOUT0	VTR	Low	
FAN_TACH0	VTR	Low	
FAN_TACH1	VTR	Low	
GPIO001	VTR	No Gate	
GPIO002	VTR	No Gate	
GPIO003	VTR	No Gate	
GPIO004	VTR	No Gate	
GPIO007	VTR	No Gate	

CEC1702 84 WFBGA	Interface		Notes
VBAT-Powered Control Interface			
B1	BGPO0	VBAT driven GPO	
B4	VCI_IN0#	Input can cause wakeup or interrupt event, active low	Note 3
A5	VCI_IN1#	Input can cause wakeup or interrupt event, active low	Note 3

2.8 Strapping Options

GPIO171 is used for the TAP Controller select strap. If any of the JTAG TAP controllers are used, GPIO171 must only be configured as an output to a VTR powered external function. GPIO171 may only be configured as an input when the JTAG TAP controllers are not needed or when an external driver does not violate the Slave Select Timing.See Section 36.2.1, "TAP Controller Select Strap Option".

TABLE 2-1: STRAPS AND MEANING

Pin	Function	Definition
GPIO171/TFDATA/ UART1_RX/(JTAG_STRAP)	JTAG Boundary Scan	1=Use the JAG TAP Controller for Boundary Scan 0=The JTAG TAP Controller is used for debug

Agg IRQ	Agg Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description	Agg NVIC	Direct NVIC
GIRQ17	0	Reserved				9	70
	1	TACH 0	TACH	No	Tachometer 0 Interrupt Event		71
	2	TACH 1	TACH	No	Tachometer 1 Interrupt Event		72
	3	Reserved					73
	4	RPM2PWM 0	FAN_FAIL	No	Failure to achieve target RPM		74
	5	RPM2PWM 0	FAN_STALL	No	Fan stall condition		75
	6	RPM2PWM 1	FAN_FAIL	No	Failure to achieve target RPM		76
	7	RPM2PWM 1	FAN_STALL	No	Fan stall condition		77
	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	Reserved					80
	11	RC-ID 1	RCID	No	0-1 transition of RC-ID done flag		81
	12	RC-ID 2	RCID	No	0-1 transition of RC-ID done flag		82
	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
	15-24	Reserved					
	25	RTOS Timer	SWI_0	No	Soft Interrupt request 0		
	26	RTOS Timer	SWI_1	No	Soft Interrupt request 1		
	27	RTOS Timer	SWI_2	No	Soft Interrupt request 2		
	28	RTOS Timer	SWI_3	No	Soft Interrupt request 3		
	29-31	Reserved					
GIRQ18	0	Reserved				10	90
	1	Quad Master SPI Controller	QMSPI_INT	No	Master SPI Controller Requires Servicing		91
	2	GP-SPI 0	TXBE_STS	No	SPI TX buffer empty		92
	3	GP-SPI 0	RXBF_STS	No	SPI RX buffer full		93
	4-31	Reserved					
GIRQ19	0-31	Reserved				11	103
GIRQ20	0-8	Test	Test	-	-	12	N/A
	9-31	Reserved					

6.7.2 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 6-4: 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)

TABLE 6-5: ARM JTAG ID

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

6.10.1 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.

6.10.2 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 6-4, "Exception and Interrupt Vector Table Layout").

6.10.2.1 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.

6.10.2.2 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.

6.10.2.3 Data Watchpoints and Trace (DWT Subblock)

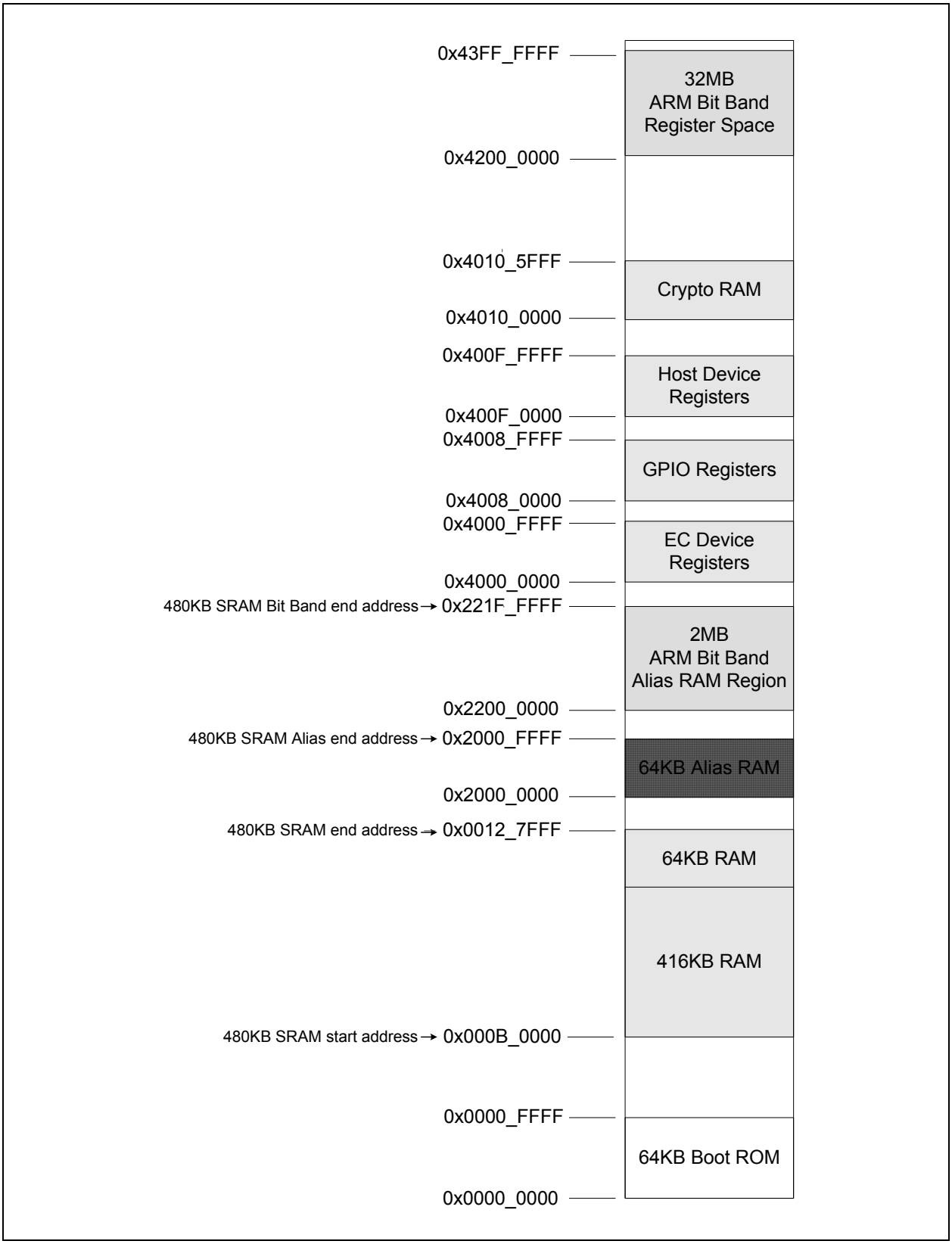
The Debug Watchpoint and Trace (DWT) block provides watchpoint support, program counter sampling for performance monitoring, and embedded trace trigger control.

6.10.2.4 Trace Interface (ETM and TPIU)

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

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

FIGURE 7-1: MEMORY LAYOUT



19.0 WEEK TIMER

19.1 Introduction

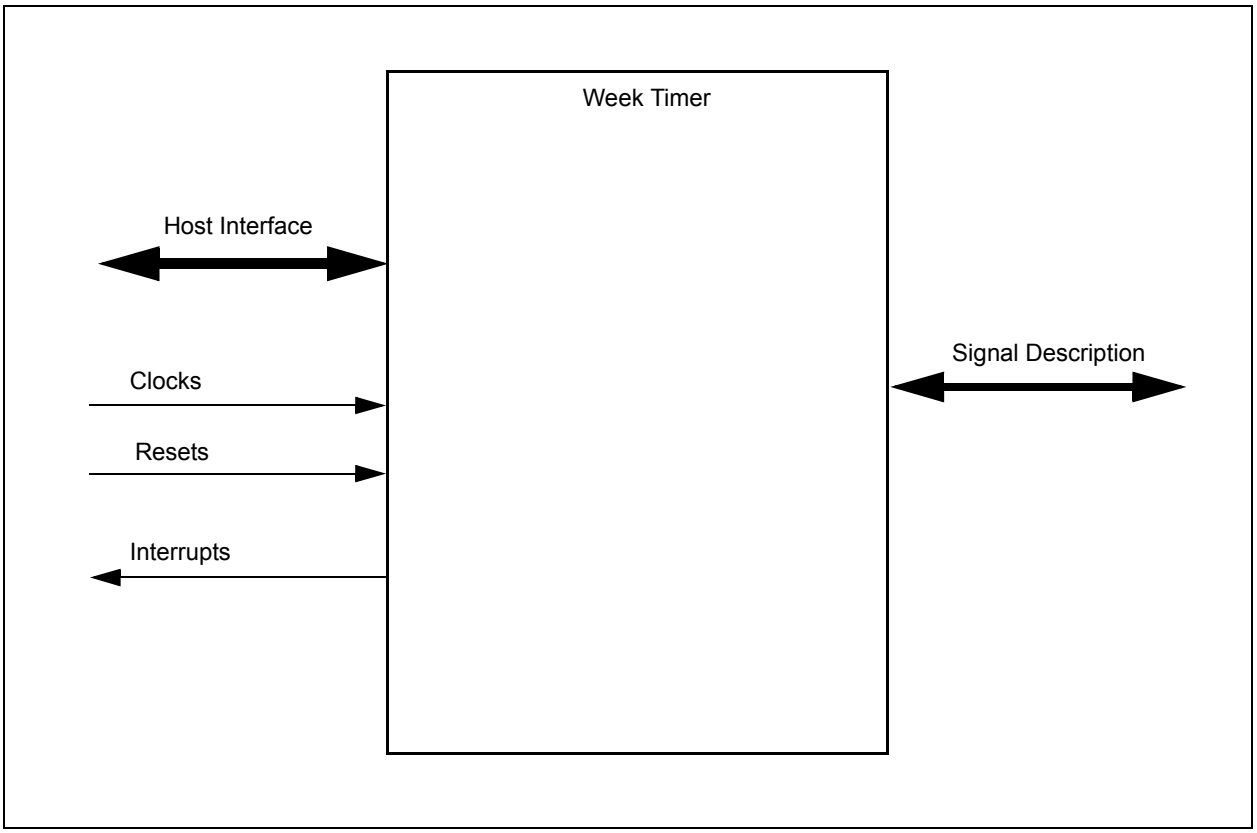
The Week Alarm Interface provides two timekeeping functions: a Week Timer and a Sub-Week Timer. Both the Week Timer and the Sub-Week Timer assert the Power-Up Event Output which automatically powers-up the system from the G3 state. Features include:

- 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

19.2 Interface

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

FIGURE 19-1: I/O DIAGRAM OF BLOCK



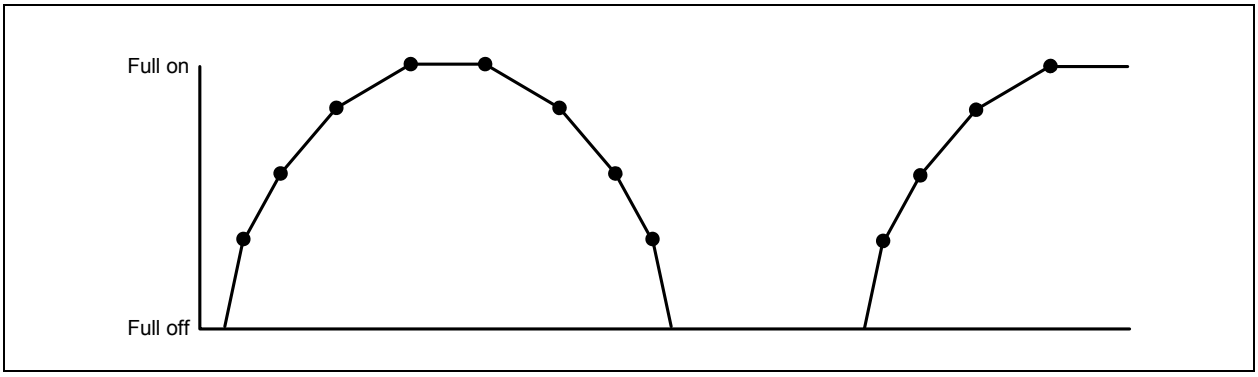
19.3 Signal Description

TABLE 19-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
BGPO[9:0]	OUTPUT	Battery-powered general purpose outputs

The brightness can also be changed in a non-linear fashion, as shown in the following figure:

FIGURE 24-4: 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.

24.8.2 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 24-2, "LED Blink Configuration Examples" shows some example blinking configurations:

TABLE 24-2: 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

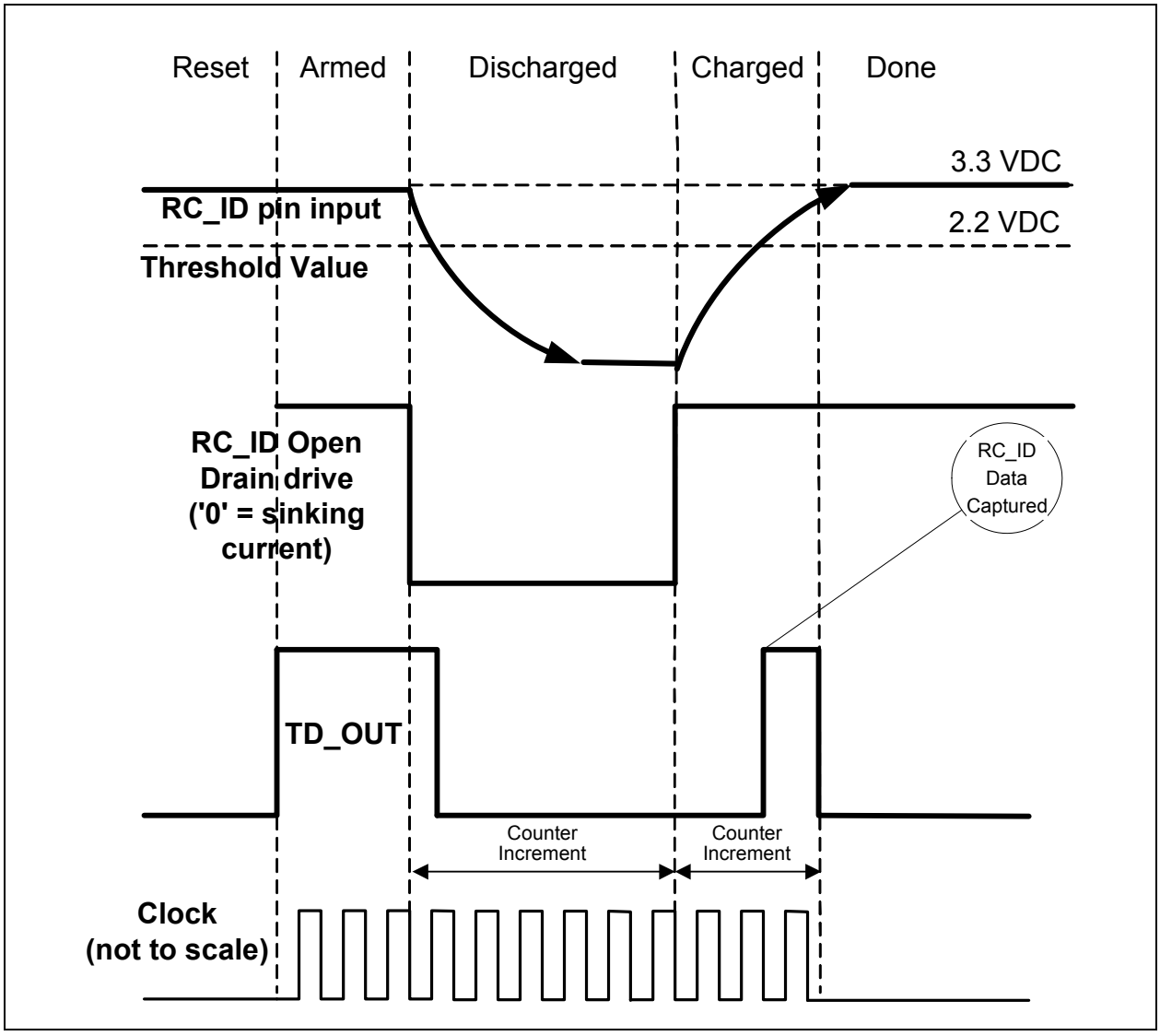
The Blinking and General Purpose PWM modes share the hardware used in the breathing mode. The Prescale value is derived from the LD field of the LED_DELAY register and the Duty Cycle is derived from the MIN field of the LED_LIMITS register.

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 25-1: 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 25-3: RCID STATE TRANSITIONS



- After 8 SPI_CLK pulses, the third SPI cycle is complete (Address Byte (LSB) transmitted):
  - EEPROM address A7-A0 has been transmitted to the slave completing the third SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit address data to the slave.
  - Once the third SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (data byte D7:D0) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- If only one data byte is to be written, the host would not write any more values to the TX_DATA register until this transaction completes. If more than one byte of data is to be written, another data byte would be written to the TX_DATA register. The SPI master automatically clears the TXFE bit when the TX_DATA register is written, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses, the fourth SPI cycle is complete (First Data Byte transmitted):
  - The data byte has been transmitted to the slave completing the fourth SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. Like the command and address phases, the data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated to transmit data to the slave.
  - Once the fourth SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (if any) and loads it into the TX shift register. This process will be repeated until all the desired data is transmitted.
- If no more data needs to be transmitted by the master, CS# and WR# are released and the SPI is idle.

28.11.2 HALF DUPLEX (BIDIRECTIONAL MODE) TRANSFER EXAMPLE

The slave device used in this example is a National LM74 12 bit (plus sign) temperature sensor.

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is asserted '1' to enable the SPI interface in Half Duplex mode.
- The CLKPOL, TCLKPH and RCLKPH bits are de-asserted '0' to match the clocking requirements of the slave device.
- The LSBF bit is de-asserted '0' to indicate that the slave expects data in MSB-first order.
- BIOEN is asserted '0' to indicate that the first data in the transaction is to be received from the slave.
- Assert CS# using a GPIO pin.

//Receive 16-bit Temperature Reading

- Write a dummy command byte (as specified by the slave device) to the SPITD - SPI TX_Data Register with TXFE asserted '1'. The SPI master automatically clears the TXFE bit indicating the byte has been put in the TX buffer. If the shift register is empty the TX_DATA byte is loaded into the shift register and the SPI master reasserts the TXFE bit. Once the data is in the shift register the SPI master begins shifting the data value onto the SPDOUT pin and drives the SPI_CLK pin. This data is lost because the output buffer is disabled. Data on the SPDIN pin is sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, another dummy byte is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the dummy data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the first receive byte
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. The data now contained in SPIRD - SPI RX_Data Register is the first half of the 16 bit word containing the temperature data.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (dummy byte 2) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock.

28.12.7 SPI CLOCK GENERATOR REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
5:0	PRELOAD SPI Clock Generator Preload value.	R/W	2h	RESET_SYS

29.0 QUAD SPI MASTER CONTROLLER

29.1 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.

29.2 References

No references have been cited for this feature.

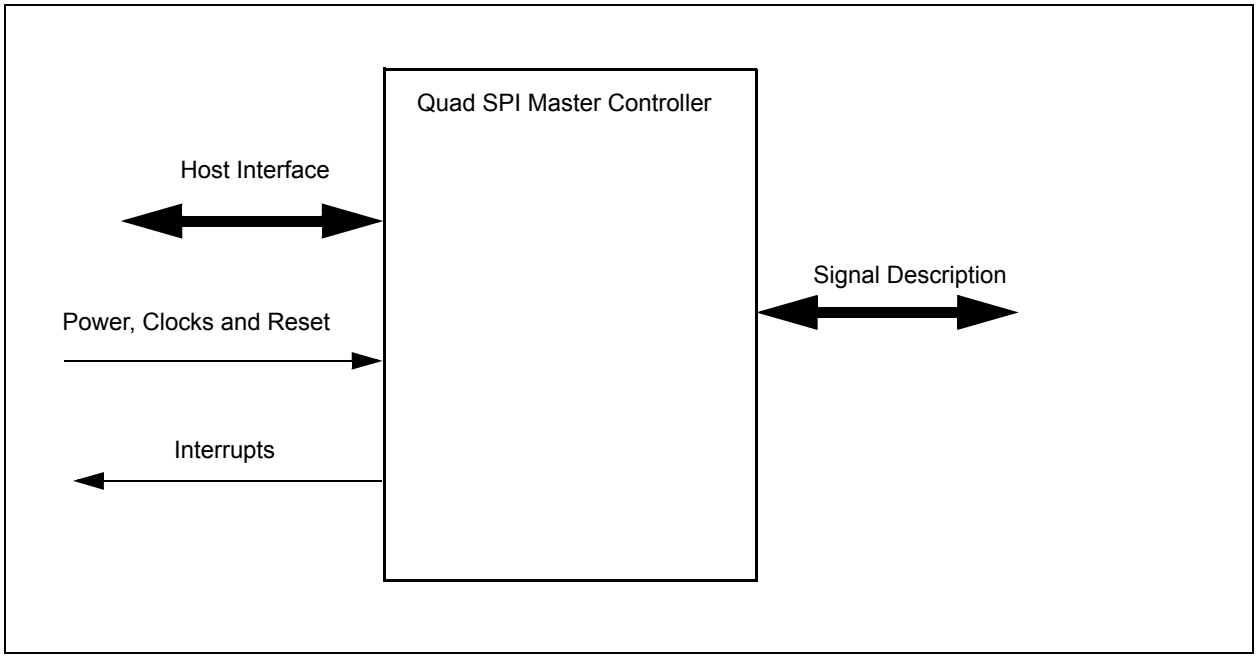
29.3 Terminology

No terminology for this block.

29.4 Interface

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

FIGURE 29-1: I/O DIAGRAM OF BLOCK



29.5 Signal Description

TABLE 29-1: EXTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
SPI_CLK	Output	SPI Clock output used to drive the SPCLK pin.
SPI_CS#	Output	SPI chip select
SPI_IO0	Input/Output	SPI Data pin 0. Also used as SPI_MOSI, Master-Out/Slave-In when the interface is used in Single wire mode
SPI_IO1	Input/Output	SPI Data pin 1. Also used as SPI_MISO, Master-In/Slave-Out when the interface is used in Single wire mode

TABLE 29-1: 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.

29.6 Host Interface

The registers defined for the General Purpose Serial Peripheral Interface are accessible by the various hosts as indicated in Section 29.11, "EC Registers".

29.7 Power, Clocks and Reset

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

29.7.1 POWER

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

29.7.2 CLOCKS

Name	Description
48MHz	This is a clock source for the SPI clock generator.

29.7.3 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.

29.8 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.

29.9 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.

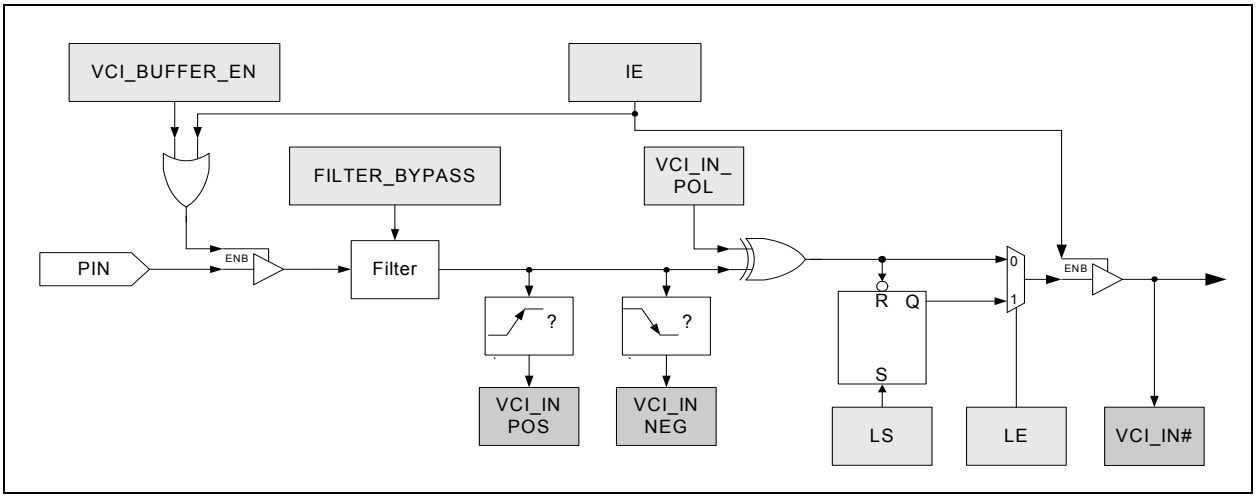
29.10 Description

- Support for multiple SPI pin configurations
  - Single wire half duplex
  - Two wire full duplex

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TRANSMIT_BUFFER_REQUEST This status is asserted if the Transmit Buffer reaches a high water mark established by the TRANSMIT_BUFFER_TRIGGER field. 1=TRANSMIT_BUFFER_COUNT is less than or equal to TRANSMIT_BUFFER_TRIGGER 0=TRANSMIT_BUFFER_COUNT is greater than TRANSMIT_BUFFER_TRIGGER	R/WC	0h	RESET
9	TRANSMIT_BUFFER_EMPTY 1=The Transmit Buffer is empty 0=The Transmit Buffer is not empty	R	0h	RESET
8	TRANSMIT_BUFFER_FULL 1=The Transmit Buffer is full 0=The Transmit Buffer is not full	R	0h	RESET
7:5	Reserved	R	-	-
4	PROGRAMMING_ERROR This bit if a programming error is detected. Programming errors are listed in Section 29.10.4, "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

The VCI_INx# Logic in the block diagram is illustrated in the following figure:

FIGURE 31-3: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



31.8.1 INPUT POLARITY

The VCI_INx# pins have an optional polarity inversion. The inversion takes place after any input filtering and before the VCI_INx# signals are latched in the VCI_INx# status bits in the VCI Register. Edge detection occurs before the polarity inversion. The inversion is controlled by battery-backed configuration bits in the VCI Polarity Register.

31.8.2 EDGE EVENT STATUS

Each VCI_INx# input pin is associated with two register bits used to record edge transitions on the pins. The edge detection takes place after any input filtering, before polarity control and occurs even if the VCI_INx# input is not enabled as part of the VCI_OUT logic (the corresponding control bit in the VCI Input Enable Register is '0') or if the state of the VCI_INx# input is not latched (the corresponding control bit in the Latch Enable Register is '0'). One bit is set whenever there is a high-to-low transition on the VCI_INx# pin (the VCI Negedge Detect Register) and the other bit is set whenever there is a low-to-high transition on the VCI_INx# pin (the VCI Posedge Detect Register).

In order to minimize power drain on the VBAT circuit, the edge detection logic operates only when the input buffer for a VCI_INx# pin is enabled. The input buffer is enabled either when the VCI_INx# pin is configured to determine the VCI_OUT pin, as controlled by the VCI_IN[1:0]# field of the VCI Register, or when the input buffer is explicitly enabled in the VCI Input Enable Register. When the pins are not enabled transitions on the pins are ignored.

31.8.3 VCI PIN MULTIPLEXING

Each of the VCI inputs, as well as VCI_OUT, are multiplexed with standard VTR-powered GPIOs. When VTR power is off, the mux control is disabled and the pin always reverts to the VCI function. The VCI_INx# function should be disabled in the VCI Input Enable Register VCI Buffer Enable Register and for any pin that is intended to be used as a GPIO rather than a VCI_INx#, so that VCI_OUT is not affected by the state of the pin.

31.8.4 POWER ON INHIBIT TIMER

The Power On Inhibit Timer prevents the VBAT-Powered Control Interface VCI_OUT pin from being asserted for a programmable time period after the SYS_QSPI0N# pin asserted. This holdoff time can be used to give a system the opportunity to cool down after a thermal shutdown before allowing a user to attempt to turn the system on. While the Inhibit Timer is asserted, the VCI_OUT pin remains de-asserted and is unaffected by the VCI, Week Alarm and RTC interfaces.

The holdoff time is configured using the Holdoff Count Register. By setting the Holdoff Count Register to 0 the Inhibit Timer is disabled. When disabled, the HOLDOFF# signal is de-asserted and no counting takes place.

The HOLDOFF# output is asserted within one 32.768KHz clock cycle from the time SYS_QSPI0N# is asserted.

33.7.2 CLOCK ENABLE REGISTER

Address	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	R	-	-
3	XOSEL This bit selects between a single-ended clock source for the crystal oscillator or an external parallel crystal. 1=The crystal oscillator is driven by a single-ended 32KHz clock source connected to the XTAL2 pin 0=The crystal oscillator requires a 32KHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins	R/W	0b	RESET_VBAT
2	32KHZ_SOURCE This field determines the source for the always-on 32KHz internal clock source. If set to '1b', this bit will only take effect if an active clock has been detected on the crystal pins. Once the 32KHz source has been switched, activity detection on the crystal no longer functions. Therefore, if the crystal oscillator uses a single-ended input, once started that input must not stop while this bit is '1b'. 1=Crystal Oscillator. The selection between a singled-ended input or a resonant crystal is determined by XOSEL in this register 0=Silicon Oscillator	R/W	0b	RESET_VBAT
1	EXT_32K This bit selects the source for the 32KHz clock domain. 1=The 32KHZ_IN VTR-powered pin is used as a source for the 32KHz clock domain. If an activity detector does not detect a clock on the selected source, the always-on 32KHz internal clock source is automatically selected 0=The always-on32Khz clock source is used as the source for the 32KHz clock domain	R/W	0b	RESET_VBAT
0	32K_SUPPRESS 1=32KHz clock domain is off while VTR is off (i.e., while on VBAT only). The 32KHz domain is always on while VTR is on, so the PLL always has a reference 0=32KHz clock domain is enabled while VTR is off (i.e., while on VBAT only). The clock source for the 32KHz domain is determined by the other bits in this register	R/W	0b	RESET_VBAT

33.7.3 MONOTONIC COUNTER REGISTER

Address	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MONOTTONIC_COUNTER Read-only register that increments by 1 every time it is read. It is reset to 0 on a VBAT Power On Reset.	R	0b	RESET_VBAT

33.7.4 COUNTER HIWORD REGISTER

Address	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER_HIWORD Thirty-two bit read/write register. If software sets this register to an incrementing value, based on an external non-volatile store, this register may be combined with the Monotonic Counter Register to form a 64-bit monotonic counter.	R/W	0b	RESET_VBAT

39.4 Clocking AC Timing Characteristics

FIGURE 39-5: CLOCK TIMING DIAGRAM

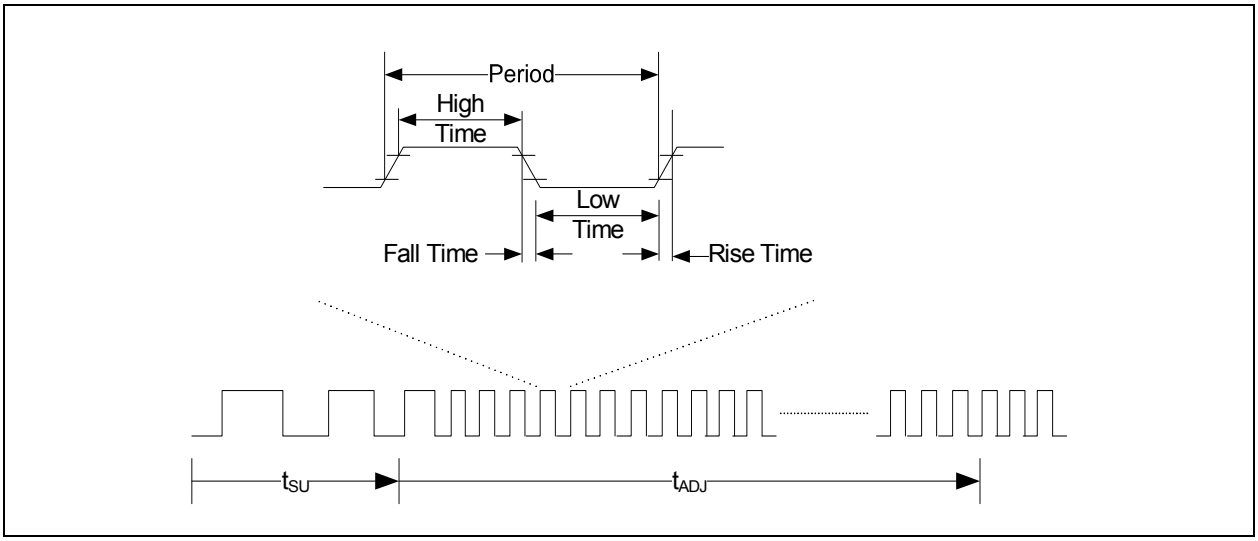


TABLE 39-4: CLOCK TIMING PARAMETERS

Clock	Symbol	Parameters	MIN	TYP	MAX	Units
48 MHz PLL	f_{SU}	Start-up accuracy	-	-	3	ms
	-	Operating Frequency (locked to 32KHz single-ended input) (Note 1)	47.5	48	48.5	MHz
	-	Operating Frequency (locked to 32KHz Silicon Oscillator) (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
32.768 kHz Crystal Oscillator (Note 3)	-	Operating Frequency	-	32.768	-	kHz

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 XTAL2 pin or 32KHZ_IN pin.

39.10 Fan Tachometer Timing

FIGURE 39-10: FAN TACHOMETER INPUT TIMING

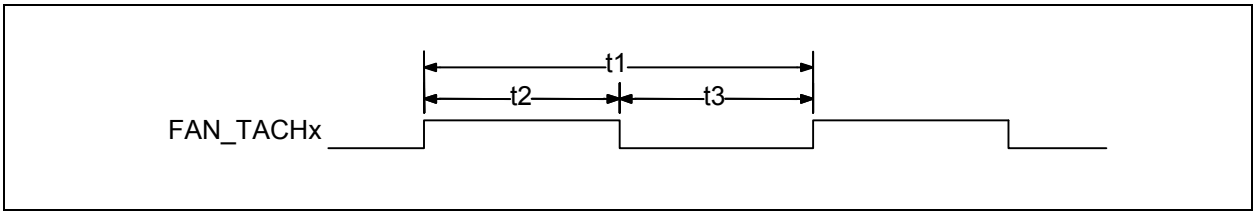


TABLE 39-10: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			µsec
t2	Pulse High Time	20			
t3	Pulse Low Time	20			
Note:	t_{TACH} is the clock used for the tachometer counter. It is 30.52 * prescaler, where the prescaler is programmed in the Fan Tachometer Timebase Prescaler register.				

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