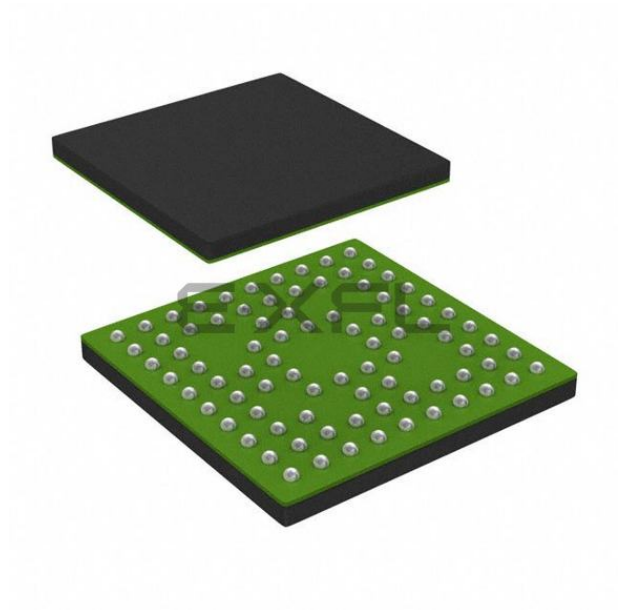


Welcome to E-XFL.COM

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	-40°C ~ 85°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-b2-i-sx

CEC1702-84	Signal	Default (if not GPIO)	Default State (if not In)
H9	GPIO107/KSO04		
G9	GPIO112/KSO05		
G10	GPIO113/KSO06		
H10	GPIO120/KSO07		
D10	GPIO121/QSPI1_IO0/KSO08		
B10	GPIO122/QSPI1_IO1/KSO09		
C10	GPIO124/QSPI1_CS#/KSO11		
A10	GPIO125/GPTP-OUT5/QSPI1_CLK/KSO12		
C6	GPIO127/UART0_CTS#		
E10	GPIO134/PWM10/UART1_RTS#		
E7	GPIO135/UART1_CTS#		
H6	GPIO140/ICT5		
C2	GPIO145/I2C09_SDA/JTAG_TDI		
B6	GPIO146/I2C09_SCL/JTAG_TDO		
A7	GPIO147/I2C08_SDA/JTAG_CLK		
B3	GPIO150/I2C08_SCL/JTAG_TMS		
A9	GPIO154/I2C02_SDA		
B7	GPIO155/I2C02_SCL		
D7	GPIO156/LED0		
B9	GPIO157/LED1		
A5	GPIO162/VCI_IN1#	VCI_IN1#	
B4	GPIO163/VCI_IN0#	VCI_IN0#	
A6	GPIO165/32KHZ_IN/CTOUT0		
F9	GPIO170/TFCLK/UART1_TX		
F8	GPIO171/TFDATA/UART1_RX/(JTAG_STRAP)		In-PU
F2	GPIO200/ADC00	ADC00	
G2	GPIO201/ADC01	ADC01	
H2	GPIO202/ADC02	ADC02	
G1	GPIO203/ADC03	ADC03	
H1	GPIO204/ADC04	ADC04	
K5	GPIO223/QSPI0_IO0		
K3	GPIO224/QSPI0_IO1		
D6	GPIO225/UART0_RTS#		
K4	GPIO227/QSPI0_IO2		
E8	JTAG_RST#	JTAG_RST#	
D2	RESETI#	RESETI#	
D5	VBAT	VBAT	
A1	VCI_OUT	VCI_OUT	O2ma-High
F1	VR_CAP	VR_CAP	
E3	VREF_ADC	VREF_ADC	
E4	VSS1	VSS1	

TABLE 5-1: POWER SOURCE DEFINITIONS (CONTINUED)

Power Well	Nominal Voltage	Description	Source
VBAT	3.0V - 3.3V	System Battery Back-up Power Well. This is the “coin-cell” battery. GPIOs that share pins with VBAT signals are powered by this supply.	Pin Interface VBAT
VSS	0V	Digital Ground	Pin Interface
Note 1: See Section 5.4.1, "I/O Rail Requirements" for connection requirements for VTRx			

5.4.1 I/O RAIL REQUIREMENTS

All pins are powered by three power supply pins: VBAT, VTR1, VTR2. The VBAT supply must be 3V to 3.6V maximum, as shown in the following section. The VTRx pins, however, may be connected to either a 3.3V or a 1.8V power supply. The device must be able to determine the voltage when the internal RESET_SYS is de-asserted in order to configure the pins properly. The device remains in reset until VTR_ANALOG has ramped to 3.3V and VTR_REG has ramped to at least 1.6V.

Software can determine whether a VTRx region is 1.8V or 3.3V by examining the GPIO Bank Power Register.

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

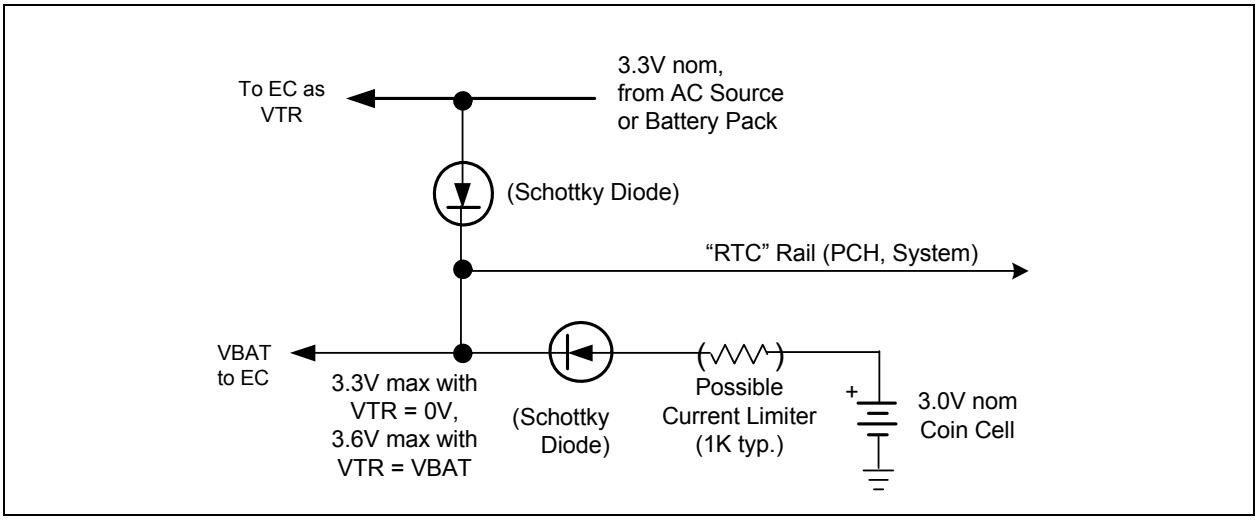
5.4.2 BATTERY CIRCUIT REQUIREMENTS

VBAT must always be present if VTR_ANALOG is present.

Microchip recommends removing all power sources to the device defined in Table 5-1, "Power Source Definitions" and all external voltage references defined in Table 5-2, "Voltage Reference Definitions" before removing and replacing the battery. In addition, upon removing the battery, ground the battery pin before replacing the battery.

The following external circuit is recommended to fulfill this requirement:

FIGURE 5-1: RECOMMENDED BATTERY CIRCUIT



5.4.3 VOLTAGE REFERENCES

Table 5-2 lists the External Voltage References to which the CEC1702 provides high impedance interfaces.

TABLE 5-2: VOLTAGE REFERENCE DEFINITIONS

Power Well	Nominal Input Voltage	Scaling Ratio	Nominal Monitored Voltage	Description	Source
VREF_ADC	Variable	n/a	Variable	ADC Reference Voltage	Pin Interface

5.5 Clocks

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

5.5.1 RAW CLOCK SOURCES

The table defines raw clocks that are either generated externally or via an internal oscillator.

TABLE 5-3: SOURCE CLOCK DEFINITIONS

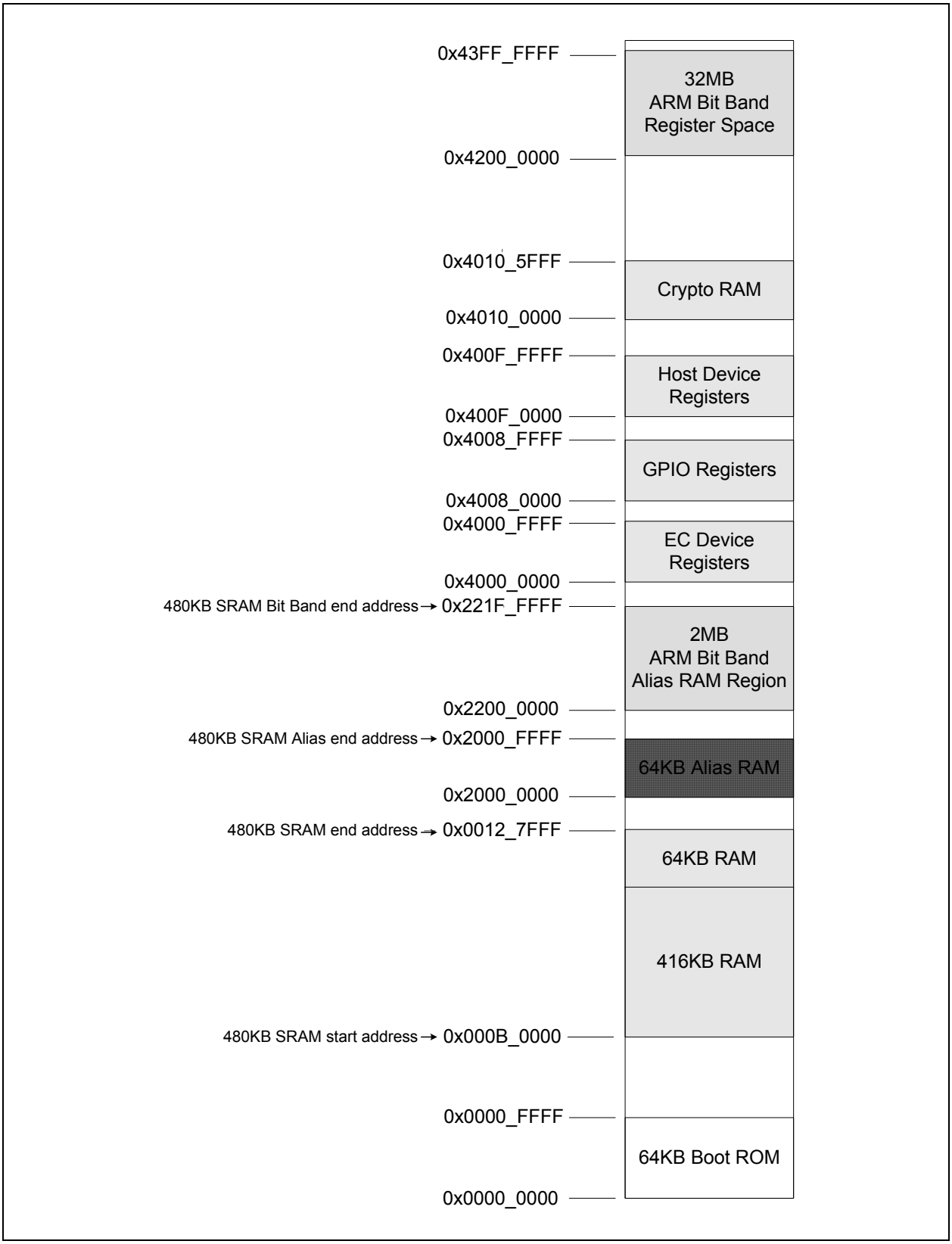
Clock Name	Frequency	Description	Source
32KHZ_IN	32.768 kHz (nominal)	Single-ended external clock input pin	32KHZ_IN pin
32.768 kHz Crystal Oscillator	32.768 kHz	A 32.768 kHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins. The accuracy of the clock depends on the accuracy of the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

5.5.2 CLOCK DOMAINS

TABLE 5-4: CLOCK DOMAIN DEFINITIONS

Clock Domain	Description
32KHz	The clock source used by internal blocks that require an always-on low speed clock
48MHz	The main clock source used by most internal blocks
96MHz	The clock source used by the Public Key Cryptographic Engine. It is derived from the same PLL as the 48MHz clock source.

FIGURE 7-1: MEMORY LAYOUT



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.

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

15.12.3 CAPTURE CONTROL 1 REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:13	FCLK_SEL5 This 3-bit field sets the clock source for the input filter for Capture Register 5. See Table 15-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
12:11	Reserved	R	-	-
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 15-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS

18.11.7 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

18.11.8 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

18.11.9 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

18.11.10 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

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	R	-	-
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.

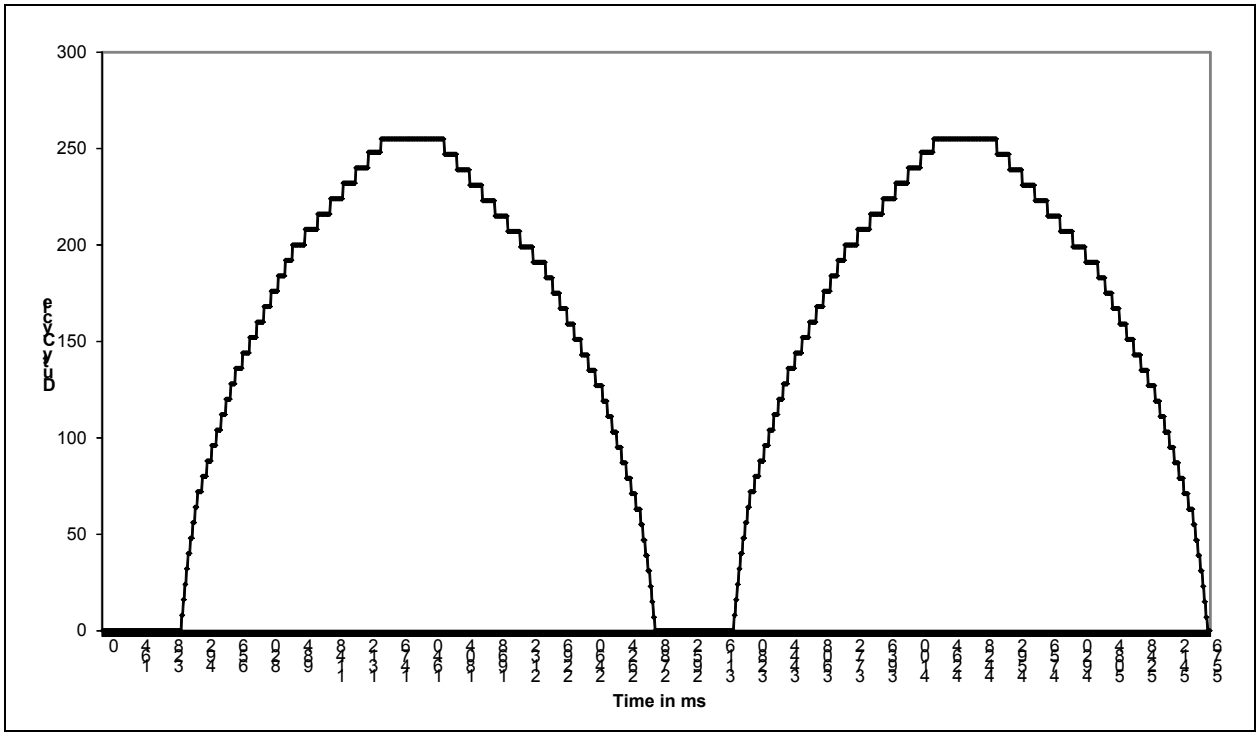
22.11.6 ADC CHANNEL READING REGISTERS

All 16 ADC channels return their results into a 32-bit reading register. In each case the low 10 bits of the reading register return the result of the Analog to Digital conversion and the upper 22 bits return 0. Table 22-6, "Register Summary" shows the addresses of all the reading registers.

Note:	The ADC Channel Reading Registers access require single 16, or 32 bit reads; i.e., two 8 bit reads will not provide data coherency.

The resulting curve is shown in the following figure:

FIGURE 24-6: NON-LINEAR BRIGHTNESS CURVE EXAMPLE



24.10 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 PWM Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 24-10: 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.

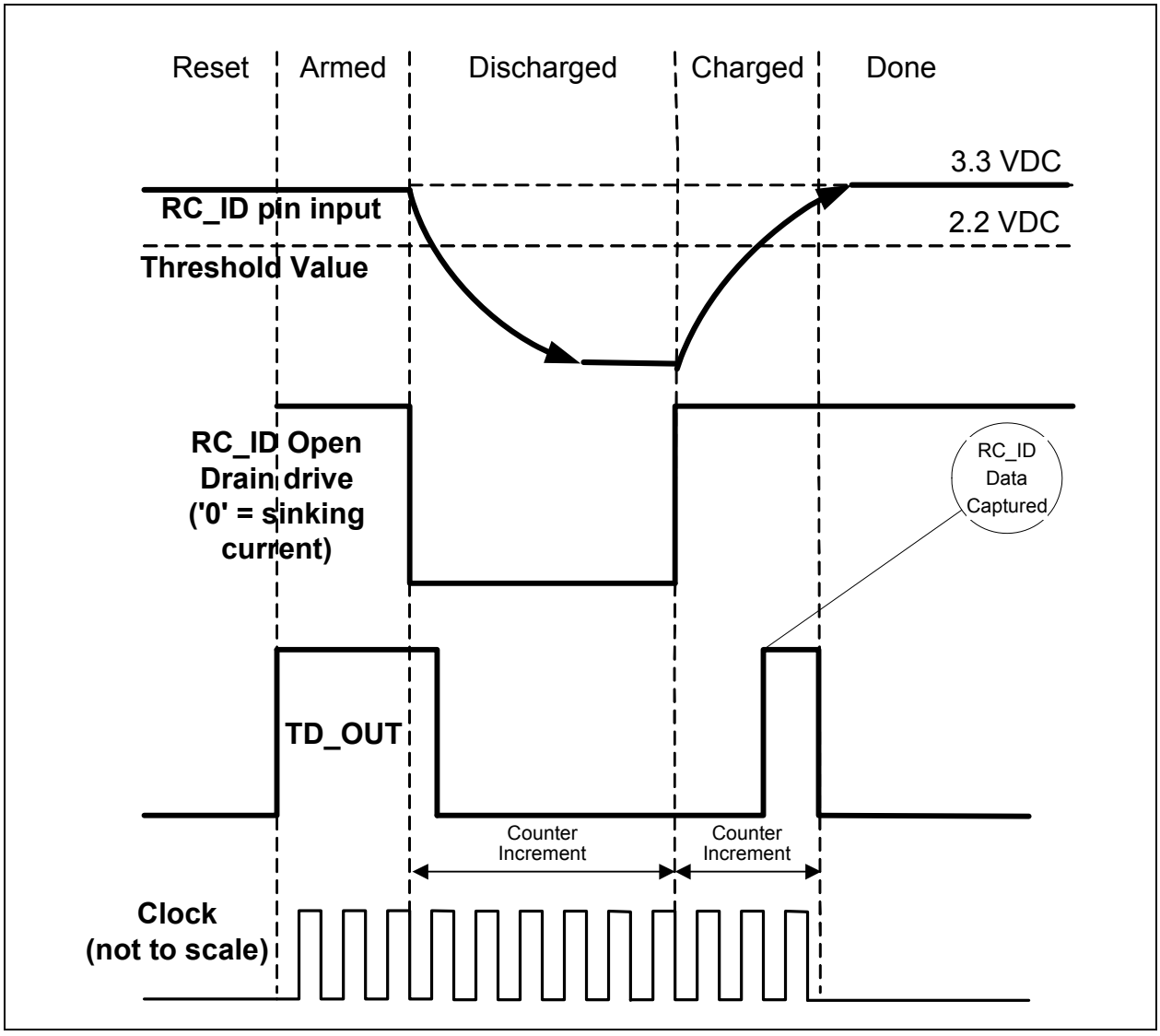
The registers in this block can be written 32-bits, 16-bits or 8-bits at a time. Writes to LED Configuration Register take effect immediately. Writes to LED Limits Register are held in a holding register and only take effect only at the end of a PWM period. The update takes place at the end of every 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-

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.

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	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
7:2	Reserved	R	-	-
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

29.11.2 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_LENGTH_BITS. 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 interface 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	1h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET

34.0 EC SUBSYSTEM REGISTERS

34.1 Introduction

This chapter defines a bank of registers associated with the EC Subsystem.

34.2 References

None

34.3 Interface

This block is designed to be accessed internally by the EC via the register interface.

34.4 Power, Clocks and Reset

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

34.4.1 POWER DOMAINS

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

34.4.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

34.4.3 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 power on.

34.5 Interrupts

This block does not generate any interrupt events.

34.6 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.

34.7 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.

34.8 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".

36.0 TEST MECHANISMS

36.1 ARM Test Functions

Test mechanisms for the ARM are described in Section 6.0, "ARM M4F Based Embedded Controller".

36.2 JTAG Boundary Scan

Note:	Boundary Scan operates in 4-wire JTAG mode only. This is not supported by 2-wire SWD.

JTAG Boundary Scan includes registers and functionality as defined in IEEE 1149.1 and the CEC1702 BSDL file. Functionality implemented beyond the standard definition is summarized in Table 36-1. The CEC1702 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.

36.2.1 TAP CONTROLLER SELECT STRAP OPTION

The TAP Controller Select Strap Option determines the JTAG slave that is selected when JTAG_RST# is not asserted. The state of the TAP Controller Select Strap Option pin, defined in the Pin Configuration chapter, is sampled by hardware at POR according to the Slave Select Timing as defined in Section 39.16, "JTAG Interface Timing" and is registered internally to select between the debug and boundary scan TAP controllers.

If the strap is sampled low, the debug TAP controller is selected; if the strap is sampled high, the boundary scan slave is selected. An internal pull-up resistor is enabled by default on the TAP Controller Select Strap Option pin and can be disabled by firmware, if necessary.

TABLE 36-1: EXTENDED BOUNDARY SCAN FUNCTIONALITY

Bits	Function	Description
12, 14	TAP Controller Select Strap Option Override	When the Strap Option Override is ‘1,’ the strap option is overridden to select the debug TAP Controller until the next time the strap is sampled. To set Strap Override Function, write 0X1FFFFD to the TAP controller instruction register, then write 0x5000 to the TAP controller data register. Note that the instruction register is 18 bits long; the data register is 16 bits long.

36.3 JTAG Master

The JTAG Master controller in the CEC1702 enables the embedded controller to perform full IEEE 1149.1 test functions as the master controller for test operations at assembly time or in the field.

The JTAG Master interface shares the JTAG pin interface with the JTAG Boundary Scan and Debug TAP controllers; including, JTAG_CLK, JTAG_TDI, JTAG_TDO and JTAG_TMS. When the CEC1702 JTAG interface is configured as master, it is the responsibility of the master firmware to satisfy all requirements regarding JTAG port multiplexing. It is also it is the responsibility of the JTAG Master firmware to satisfy all requirements for external JTAG slave devices that require an external asynchronous reset (TRST#) input.

When JTAG slave functions are not required and the JTAG Master is enabled, the JTAG Interface pins are turned around so that the pins JTAG_CLK, JTAG_TMS and JTAG_TDI become outputs and the JTAG_TDO becomes an input.

Figure 36-1, "JTAG Signal Clocking" shows the clocking behavior of JTAG in the TAP controller in a JTAG Slave device. The rows “TAP State” and “Shift Reg. Contents” refer to the state of the JTAG Slave device and are provided for reference. When configured as a Master, the JTAG interface drives JTAG_CLK and will shift out data onto JTAG_TMS and JTAG_TDI in parallel, updating the pins on the falling edge of JTAG_CLK. The Master will sample data on JTAG_TDO on the rising edge of JTAG_CLK.

38.2.4 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 38-3: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO Type Buffer						
All PIO Buffers						Internal PU/PD selected via the GPIO Pin Control Register.
Pull-up current	I_{PU}	39	84	162	KΩ	
Pull-down current	I_{PD}	39	65	105	KΩ	
PIO						The drive strength is determined by programming bits[5:4] of the Pin Control 2 Register
DRIVE_STRENGTH = 00b	_	_	_	_	_	Same characteristics as an IO-2 mA.
DRIVE_STRENGTH = 01b	_	_	_	_	_	Same characteristics as an IO-4 mA.
DRIVE_STRENGTH = 10b	_	_	_	_	_	Same characteristics as an IO-8 mA.
DRIVE_STRENGTH = 11b	_	_	_	_	_	Same characteristics as an IO-12 mA.
I Type Input Buffer						TTL Compatible Schmitt Trigger Input
Low Input Level	V_{ILI}			0.3x VTR	V	
High Input Level	V_{IHI}	0.7x VTR			V	
Schmitt Trigger Hysteresis	V_{HYS}		400		mV	
O-2 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 2 mA
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -2 mA
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
IOD-2 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-2mA.

TABLE 39-19: 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.

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