Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	-
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, KB, LED, POR, PWM, WDT, Crypto - AES, SHA, RSA, TRNG
Number of I/O	116
Program Memory Size	-
Program Memory Type	External Program Memory
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	3.135V ~ 3.465V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	144-WFBGA
Supplier Device Package	144-WFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/cec1302d-c0-sz

Note: Table 1-1, "CEC1302 144 WFBGA Pin Configuration" shows the mapping between Pin Ref. Number and 144 WFBGA ball number.

TABLE 1-1: CEC1302 144 WFBGA PIN CONFIGURATION

Pin Ref. Number	144 WFBGA Number	Pin Name
1	C3	GPIO036
2	F5	GPIO153/PVT_SCLK
3	F6	GPIO122/SHD_SCLK
4	A2	GPIO011/KSO16
5	A1	KSO13/GPIO006
6	B1	KSO12/GPIO005
7	B2	KSO11/GPIO107
8	C2	KSO10/GPIO004
9	C1	KSO09/GPIO106
10	D2	KSO08/GPIO003
11	D1	VSS
12	E2	KSO07/GPIO002
13	E1	KSO06/GPIO001
14	G5	VCC1
15	F1	CAP
16	G2	KSO05/GPIO104/TFDP_CLK
17	H3	KSO04/GPIO103/TFDP_DATA/XNOR
18	H1	KSO03/GPIO102/JTAG_TDO
19	J1	KSO02/GPIO101/JTAG_TDI
20	H2	KSO01/GPIO100/JTAG_TMS
21	J2	KSO00/GPIO000/JTAG_TCK
22	K1	KSI7/GPIO043
23	K3	KSI6/GPIO042
24	K2	KSI5/GPIO040
25	L1	KSI4/GPIO142/TRACECLK
26	L2	KSI3/GPIO032/TRACEDATA0
27	L3	KSI2/GPIO144/TRACEDATA1
28	M2	KSI1/GPIO126/TRACEDATA2
29	M1	KSI0/GPIO125/TRACEDATA3
30	N2	GPIO031
31	N1	GPIO127
32	M3	Reserved/GPIO052
33	N3	GPIO147
34	M4	GPIO151
35	L4	Reserved/GPIO051
36	E3	VSS
37	H5	VCC1
38	N5	ADC4/GPIO062
39	M5	ADC3/GPIO061
40	L5	AVCC
41	N6	GPIO206
42	M6	ADC2/GPIO060
43	L6	ADC1/GPIO057
44	N7	ADC0/GPIO056
45	M7	AVSS
46	N8	Reserved/GPIO112
47	A5	VSS
48	M8	Reserved/GPIO114
49	J3	JTAG_RST#
50	L8	Reserved/GPIO113
51	L9	Reserved/GPIO111
52	N9	Reserved/GPIO120
53	N10	LRESET#/GPIO116
54	M9	Reserved/GPIO117
55	M10	Reserved/GPIO014
56	F3	VSS
57	L10	Reserved/GPIO115
58	J5	VCC1
59	N11	Reserved/GPIO041
60	N12	nRESET_OUT/GPIO121
61	N13	Reserved/GPIO050
62	L11	Reserved/GPIO065
63	M12	GPIO035
64	M13	GPIO027
65	L12	GPIO033
66	K11	Reserved/GPIO046
67	J12	Reserved/GPIO047
68	K12	VBAT
69	L13	XTAL2
70	K13	VSS_VBAT
71	J13	XTAL1
72	J11	VCC_PWRGD/GPIO063

1.4.7 GENERAL PURPOSE I/O INTERFACE

TABLE 1-9: GPIO INTERFACE

GPIO Interface			
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
See Pin Configuration Table	GPIO	General Purpose Input Output Pins	Note 12

Note: No GPIO pin should be left floating in a system. If a GPIO pin is not in use, it should be either tied high, tied low, or pulled to either power or ground through a resistor.

1.4.8 MISCELLANEOUS FUNCTIONS

TABLE 1-10: MISCELLANEOUS FUNCTIONS

MISC Functions			(12 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
113	LED0	LED (Bllinking/Breathing PWM) Output 0	
114	LED1	LED (Bllinking/Breathing PWM) Output 1	
115	LED2	LED (Bllinking/Breathing PWM) Output 2	
78	LED3	LED (Bllinking/Breathing PWM) Output 3	
16	TFDP_CLK	Trace FIFO debug port - clock	
17	TFDP_DATA	Trace FIFO debug port - data	
60	nRESET_OUT	EC-driven External System Reset	Note 6
72	VCC_PWRGD	System Main Power Indication	
77	VCC1_RST#	Reset Generator Output	
85	RSMRST#	Resume Reset Output	Note 6
17	XNOR	Test Output	
53	LRESET#	Reset Signal	Note 1

Note: See Section 1.6, "Notes for Tables in this Chapter," on page 36 for numbered notes.

a) The nRESET_OUT pin function is an external output signal version of the internal signal nSIO_RESET. See the iRESET_OUT bit in the Power Reset Control (PWR_RST_CTRL) Register on page 62 and nSIO_RESET in Table 3-6, “Definition of Reset Signals,” on page 44.

b) XNOR is a push-pull output. This function is not configured through the associated GPIO Pin Control Register; however the drive strength is configured through the associated GPIO Pin Control Register 2.

c) The Resume Reset Output (RSMRST#) pin drives low as a push-pull output following a VCC1 power-on until firmware reconfigures the GPIO143 control register. This pin may be used to hold the system in reset until the CEC1302 firmware is ready to release it.

d) The LRESET# system reset pin requires an external weak pull-up resistor to VCC1 of 10k-100k ohms. See Note 1 in Section 1.6, "Notes for Tables in this Chapter," on page 36.

TABLE 1-34: MULTIPLEXING TABLE (18 OF 18)

Pin Ref. Number	MUX	Signal	Buffer Type	Default Operation	Signal Power Well	Emulated Power Well	Gated State	Notes
137	Default: 0	GPIO201	PIO	I (PD)	VCC1	ON	No Gate	
137	1	Reserved	Reserved		Reserved	Reserved		
137	2	Reserved	Reserved		Reserved	Reserved		
137	3	Reserved	Reserved		Reserved	Reserved		
138		VSS	PWR		PWR	PWR		
138								
138								
138								
139	Default: 0	GPIO203	PIO	I (PD)	VCC1	ON	No Gate	
139	1	Reserved	Reserved		Reserved	Reserved		
139	2	Reserved	Reserved		Reserved	Reserved		
139	3	Reserved	Reserved		Reserved	Reserved		
140		VSS	PWR		PWR	PWR		
140								
140								
140								
141	Default: 0	GPIO204	PIO	I (PD)	VCC1	ON	No Gate	
141	1	Reserved	Reserved		Reserved	Reserved		
141	2	Reserved	Reserved		Reserved	Reserved		
141	3	Reserved	Reserved		Reserved	Reserved		
142		NC						
142								
142								
142								
143		VSS	PWR		PWR	PWR		
143								
143								
143								
144		VSS	PWR		PWR	PWR		
144								
144								
144								

1.6 Notes for Tables in this Chapter

Note 1	The LRESET# pin requires an external weak pull-up resistor to VCC1 of 10k-100k ohms. If the LRESET# pin is assigned to the GPIO function rather than LRESET#, the internal LRESET# signal is gated low, and therefore the nRESET_OUT function, the UART and the GPIO blocks will not operate properly.
Note 2	When the JTAG_RST# pin is not asserted (logic '1'), the JTAG_TDI, JTAG_TDO, JTAG_TCK, JTAG_TMS signal functions in the JTAG interface are unconditionally routed to the interface; the Pin Control register for these pins has no effect. When the JTAG_RST# pin is asserted (logic '0'), the JTAG_TDI, JTAG_TDO, JTAG_TCK, JTAG_TMS signal functions in the JTAG interface are not routed to the interface and the Pin Control Register for these pins controls the muxing. The pin control registers can not be used to route the JTAG interface to the pins. The System Board Designer should terminate this pin in all functional states using jumpers and pull-up or pull down resistors, etc.
Note 3	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.
Note 4	A pull-down is required on the GPIO146/PVT_CS0# pin if there is no private SPI flash device on the board.
Note 5	This I2C port supports 1Mbps (pin 88, GPIO023/I2C1_DAT0 and pin 89, GPIO022/I2C1_CLK0). For 1Mbps 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.
Note 6	The following glitch protected pins require a pull-down on the board: pin 60, nRESET_OUT/GPIO121 and pin 85, GPIO143/RSMRST#. The nRESET_OUT pin will drive low when VCC1 comes on and stays low until the iRESET_OUT bit is cleared after VCC PWRGD asserts. The RSMRST# pin also drives low (as a GPIO push-pull output) following a VCC1 power-on until firmware deasserts it by writing the GPIO data bit to '1'. The GPIO143/RSMRST# pin operates in this manner as a GPIO; the RSMRST# function is not a true alternate function and the GPIO143 control register must not be changed from the GPIO default function.
Note 7	The BC DAT pin requires a weak pull up resistor (100 K Ohms).
Note 8	The voltage on the ADC pins must not exceed 3.6 V or damage to the device will occur.
Note 9	The XTAL1 pin should be left floating when using the XTAL2 pin for the single ended clock input.
Note 10	The SPI pins are configured to their SPI function by ROM boot code as follows. Shared SPI pins are configured to the following SPI functions: SHD_CLK, SHD_MOSI, SHD_MISO and SHD_CS0#. If the PVT_CS0# pin (pin 96) is sampled high, then the private SPI pins are configured to the following SPI functions after a successful load from flash: PVT_CLK, PVT_MOSI, PVT_MISO and PVT_CS0#; otherwise these pins are left as the GPIO function. It is recommended that user code reconfigures the shared SPI pins to the GPIO input function before releasing RSMRST#.
Note 11	The KSI[7:0] pins have the internal pull-up enabled by ROM boot code. Therefore the Buffer Type on these pins is I (PU) after the ROM boot code runs.

1.7 Pin States After VCC1 Power-On

Pins that default to IOD or OD in the Multiplexing Tables are open drain and come up tri-stated after VCC1 power-on. Pins that default to I are inputs and also come up tri-stated (high-z).

Table 1-35 shows pins that have specific states after VCC1 power-on.

Offset	34h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	VCC Reset Status Indicates the status of PWRGD. 0 = reset active (PWRGD not asserted). 1 = reset not active (PWRGD asserted).	R	xh	Note 3-13
1:0	RESERVED	RES		

Note 3-13 This read-only status bit always reflects the current status of the event and is not affected by any Reset events.

3.9.14 CHIP RESET ENABLE REGISTER (CHIP_RST_EN)

Offset	38h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	RESERVED	RES		
1	MCHP Reserved	R	0h	VCC1_RESET
0	MCHP Reserved	R/W	0h	VCC1_RESET

Note: If a block is configured such that it is to be reset when it goes to sleep, then registers within the block may not be writable when the block is asleep.

3.9.15 HOST RESET ENABLE REGISTER (HOST_RST_EN)

Offset	3Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	RESERVED	RES		
18	RTC Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	VCC1_RESET
17	RESERVED	RES		
16:12	MCHP Reserved	RES	0h	VCC1_RESET
11:2	RESERVED	RES		
1	UART 0 Reset Enable 0: block will not be reset on sleep. 1: block will be reset on sleep.	R/W	0h	VCC1_RESET
0	MCHP Reserved	R/W	0h	VCC1_RESET

Note: If a block is configured such that it is to be reset when it goes to sleep, then registers within the block may not be writable when the block is asleep.

6.0 ARM M4F BASED EMBEDDED CONTROLLER

6.1 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 IP has configurable options, which are selected as follows.

- **Little-Endian** byte ordering is selected at all times (hard-wired)
- **Bit Banding** feature 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
- **FPB** Unit provides HW Breakpointing with 6 Instruction and 2 Literal (Read-Only Data) address comparators. The FPB comparators are also available for Patching: remapping Instruction and Literal Data addresses.
- **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 currently 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 currently included**.
- Microchip Interrupt Aggregator function (at chip level) is expected to provide Wake control instead.
- The ARM-defined **MPU** feature is **not currently included**.
- Memory Protection functionality is not expected to be necessary.

6.2 References

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

9.4.1 SIGNAL INTERFACE

This block is not accessible from the pin interface.

9.4.2 HOST INTERFACE

The registers defined for the EC Interrupt Aggregator are only accessible by the embedded controller via the EC-Only Registers.

9.5 Power, Clocks and Reset

9.5.1 BLOCK POWER DOMAIN

TABLE 9-1: BLOCK POWER

Power Well Source	Effect on Block
VCC1	The EC Interrupt Aggregator block and registers operate on this single power well.

9.5.2 BLOCK CLOCKS

None

9.5.3 BLOCK RESET

TABLE 9-2: BLOCK RESETS

Reset Name	Reset Description
VCC1_RESET	This signal is used to indicate when the VCC1 logic and registers in this block are reset.

9.6 Interrupts

This block aggregates all the interrupts targeted for the embedded controller into the Source Registers defined in Section 9.9, "EC-Only Registers," on page 105. 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.

9.7 Low Power Modes

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

9.8 Description

The interrupt generation logic is made of 16 groups of signals, each of which consist of a Status register, a Enable register and a Result register.

The Status and Enable are latched registers. 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. See Chapter 27.0, "EC Subsystem Registers"

Section 9.8.1 shows a representation of the interrupt structure.

9.8.2 INTERRUPT SUMMARY

Table 9-3, "Interrupt Event Aggregator Routing Summary" summarizes the interrupts, wake capabilities and NVIC vector locations.

Table 9-4, "EC Interrupt Structure" summarizes the interrupts, priorities and vector locations.

TABLE 9-3: INTERRUPT EVENT AGGREGATOR ROUTING SUMMARY

Interrupt	Aggregator IRQ	Aggregator Bit	Wake Event	Aggregated NVIC	Direct NVIC Interrupt
GPIO140	GIRQ8	0	Yes	57	N/A
GPIO141	GIRQ8	1			
GPIO142	GIRQ8	2			
GPIO143	GIRQ8	3			
GPIO144	GIRQ8	4			
GPIO145	GIRQ8	5			
GPIO146	GIRQ8	6			
GPIO147	GIRQ8	7			
GPIO150	GIRQ8	8			
GPIO151	GIRQ8	9			
GPIO152	GIRQ8	10			
GPIO153	GIRQ8	11			
GPIO154	GIRQ8	12			
GPIO155	GIRQ8	13			
GPIO156	GIRQ8	14			
GPIO157	GIRQ8	15			
GPIO160	GIRQ8	16			
GPIO161	GIRQ8	17			
GPIO162	GIRQ8	18			
GPIO163	GIRQ8	19			
GPIO164	GIRQ8	20			
GPIO165	GIRQ8	21			

TABLE 9-15: BIT DEFINITIONS FOR GIRQ12 SOURCE, ENABLE, AND RESULT REGISTERS

Bit	Block Instance Name	Source Name	Wake	Source Description
3	I2C3 / SMB3	SMB	N	I2C/SMBus controller 3 interrupt. This interrupt is signaled when the I2C/SMBus controller 3 asserts its interrupt request.
4	I2C0_0_WK	SMB	Y	I2C/SMBus controller 0 (port 0) Wake interrupt. This interrupt is signaled when there is activity on the I2C/SMBus controller 0 port 0 data pin, I2C0_DAT0 (see Note 9-2 on page 118).
5	I2C0_1_WK	SMB	Y	I2C/SMBus controller 0 (port 1) Wake interrupt. This interrupt is signaled when there is activity on the I2C/SMBus controller 0 port 1 data pin, I2C0_DAT1 (see Note 9-2 on page 118).
6	I2C2_0_WK	SMB	Y	I2C/SMBus controller 2 (port 0) Wake interrupt. This interrupt is signaled when there is activity on the I2C/SMBus controller 2 (port 0) data pin, I2C2_DAT0 (see Note 9-2 on page 118).
7	I2C1_0_WK	SMB	Y	I2C/SMBus controller 1 (port 0) Wake interrupt. This interrupt is signaled when there is activity on the I2C/SMBus controller 1 port 0 data pin, I2C1_DAT0 (see Note 9-2 on page 118).
8	I2C3_0_WK	SMB	Y	I2C/SMBus controller 3 (port 0) Wake interrupt. This interrupt is signaled when there is activity on the I2C/SMBus controller 3 port 0 data pin, I2C3_DAT0 (see Note 9-2 on page 118).
[30:9]	Reserved	Reserved	N	Reserved
31	n/a	n/a	N	See Table 9-7, "GIRQx Source Register", Table 9-8, "GIRQx Enable Set Register", Table 9-10, "GIRQx Enable Clear Register", and Table 9-9, "GIRQx Result Register" for a definition of this bit for the Source, Enable, and Result registers.

9.9.6 GIRQ13

TABLE 9-16: BIT DEFINITIONS FOR GIRQ13 SOURCE, ENABLE, AND RESULT REGISTERS

Bit	Block Instance Name	Source Name	Wake	Source Description
[15:0]	Reserved	Reserved	N	Reserved
16	IRQ_DMA0	DMA0	N	Direct Memory Access Channel 0
17	IRQ_DMA1	DMA1	N	Direct Memory Access Channel 1
18	IRQ_DMA2	DMA2	N	Direct Memory Access Channel 2
19	IRQ_DMA3	DMA3	N	Direct Memory Access Channel 3
20	IRQ_DMA4	DMA4	N	Direct Memory Access Channel 4
21	IRQ_DMA5	DMA5	N	Direct Memory Access Channel 5
22	IRQ_DMA6	DMA6	N	Direct Memory Access Channel 6
23	IRQ_DMA7	DMA7	N	Direct Memory Access Channel 7
24	IRQ_DMA8	DMA8	N	Direct Memory Access Channel 8
25	IRQ_DMA9	DMA9	N	Direct Memory Access Channel 9
26	IRQ_DMA10	DMA10	N	Direct Memory Access Channel 10
27	IRQ_DMA11	DMA11	N	Direct Memory Access Channel 11

14.3 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 14-4: INTERRUPTS

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 ensured to generate an interrupt/wakeup event is 5ns.

14.4 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.

- Output GPIO Data
  - Outputs to individual GPIO ports are grouped into 32-bit GPIO Output Registers.
- Alternative GPIO 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.

14.5 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 CEC1302 GPIO indexing is done sequentially starting from 'GPIO000.'

14.6 GPIO Multiplexing Control

Pin multiplexing depends upon the Mux Control bits in the Pin Control Register. There are two Pin Control Registers for each GPIO signal function.

The CEC1302 Pin Control Register address offsets shown in the following tables depends on the GPIO Index number. Pin Control Register defaults are also shown in these tables.

Note 1: Pin Control Register 2 default values are not shown in these tables.

2: The GPIO143/RSMRST# pin operates as described in Section 1.6, "Notes for Tables in this Chapter," on page 36 when it is configured as a GPIO; the RSMRST# function is not a true alternate function. For proper RSMRST# operation on the pin, the GPIO143 control register must not be changed from the GPIO default function.

3: The VCC1_RST#/GPIO131 pin cannot be used as a GPIO pin. The input path to the VCC1_RST# logic is always active and will cause a reset if this pin is set low in GPIO mode.

4: The KSI[7:0] pins have the internal pullups enabled by ROM boot code. Therefore the Pin Control Reg. POR Value is as follows after the ROM boot code runs:

GPIO043 = 00003001h
GPIO042 = 00003001h

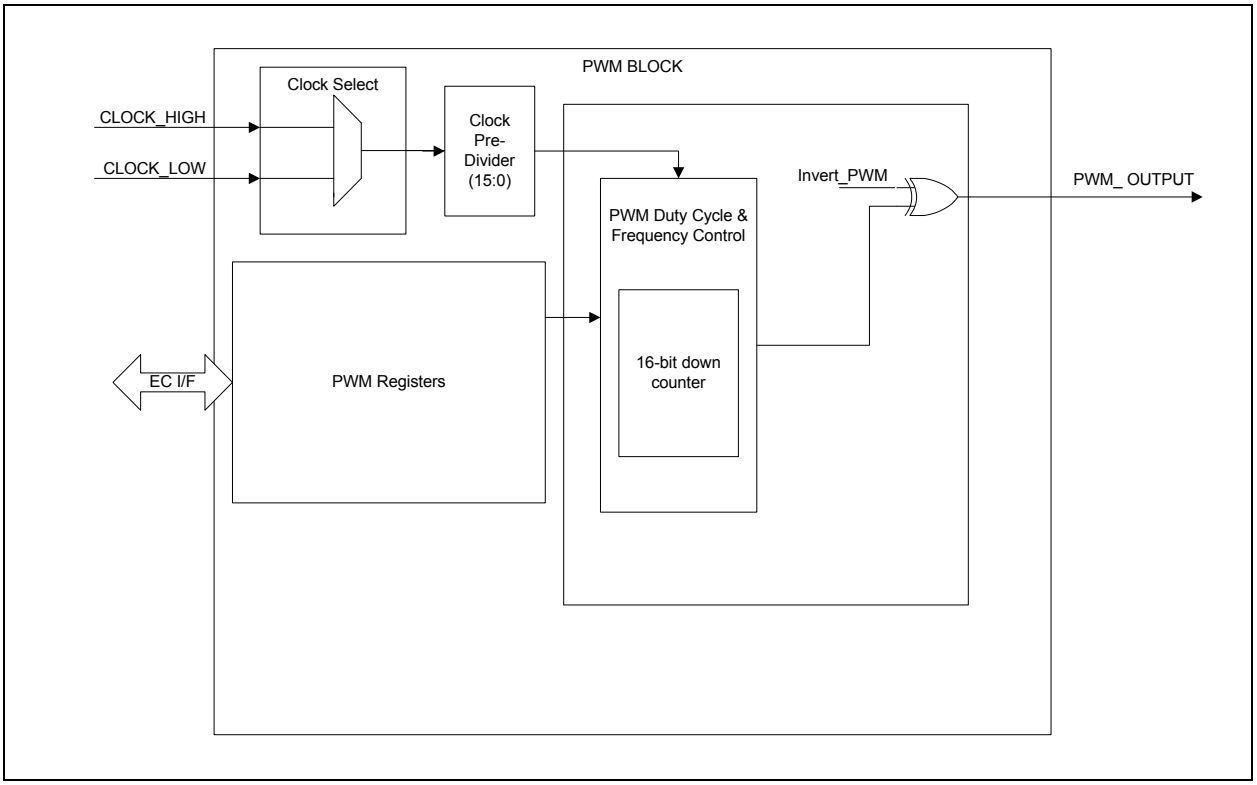
17.11.3 TACHX HIGH LIMIT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	-	-	-
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	VCC1_RESET

17.11.4 TACHX LOW LIMIT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
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	VCC1_RESET

FIGURE 18-2: BLOCK DIAGRAM OF PWM CONTROLLER



Note: In Figure 18-2, the 48 MHz Ring Oscillator is represented as CLOCK_HIGH and 100kHz_Clk 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.

FIGURE 18-3: PWM FREQUENCY EQUATION

$$\text{PWM Frequency} = \frac{1}{(PreDivisor + 1)} \times \frac{(ClockSourceFrequency)}{(PWMCounterOnTime + PWMCounterOffTime)}$$

In Figure 18-3, 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 18.11, "EC-Only Registers".

DMA controller then reads the received byte from the GP-SPI's SPI RX Data Register (SPIRD) and stores it in the DMA receive buffer. With AUTO_READ set, this read clears both the RxBF and TxBE. Clearing TxBE causes (dummy) data from the SPI TX Data Register (SPITD) to be transferred to the internal shift register, mimicking the effect of the aforementioned write to the SPI TX Data Register (SPITD) by the driver. SPI clock is toggled again to shift in the second read byte. This process continues until the end of the DMA buffer is reached - the DMA controller stops responding to an active Tx Empty until the buffer's address registers are reprogrammed.

20.10.3 TYPES OF SPI TRANSACTIONS

The GP-SPI controller can be configured to operate in three modes: Full Duplex, Half Duplex, and Dual Mode.

20.10.3.1 Full Duplex

In Full Duplex Mode, serial data is transmitted and received simultaneously by the SPI master over the SPDOUT and SPDIN pins. To enable Full Duplex Mode clear SPDIN Select.

When a transaction is completed in the full-duplex mode, the RX_DATA shift register always contains received data (valid or not) from the last transaction.

20.10.3.2 Half Duplex

In Half Duplex Mode, serial data is transmitted and received sequentially over a single data line (referred to as the SPDOUT pin). To enable Half Duplex Mode set SPDIN Select to 01b. The direction of the SPDOUT signal is determined by the BIOEN bit.

- To transmit data in half duplex mode set the BIOEN bit before writing the TX_DATA register.
- To receive data in half duplex mode clear the BIOEN bit before writing the TX_DATA register with a dummy byte.

Note:	The Software driver must properly drive the BIOEN bit and store received data depending on the transaction format of the specific slave device.

20.10.3.3 Dual Mode of Operation

In Dual Mode, serial data is transmitted sequentially from the SPDOUT pin and received in by the SPI master from the SPDOUT and SPDIN pins. This essentially doubles the received data rate and is often available in SPI Flash devices. To enable Dual Mode of operation the SPI core must be configured to receive data in path on the SPDIN1 and SPDIN2 inputs via SPDIN Select. The BIOEN bit determines if the SPI core is transmitting or receiving. The setting of this bit determines the direction of the SPDOUT signal. The SPDIN Select bits are configuration bits that remain static for the duration of a dual read command. The BIOEN bit must be toggled to indicate when the SPI core is transmitting and receiving.

- To transmit data in dual mode set the BIOEN bit before writing the TX_DATA register.
- To receive data in dual mode clear the BIOEN bit before writing the TX_DATA register with a dummy byte. The even bits (0,2,4,and 6) are received on the SPDOUT pin and the odd bits (1,3,5,and 7) are received on the SPDIN pin. The hardware assembles these received bits into a single byte and loads them into the RX_DATA register accordingly.

The following diagram illustrates a Dual Fast Read Command that is supported by some SPI Flash devices.

21.0 BLINKING/BREATHING PWM

21.1 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 48 MHz clock or by a 32.768 KHz clock input. When driven by the 48 MHz 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.

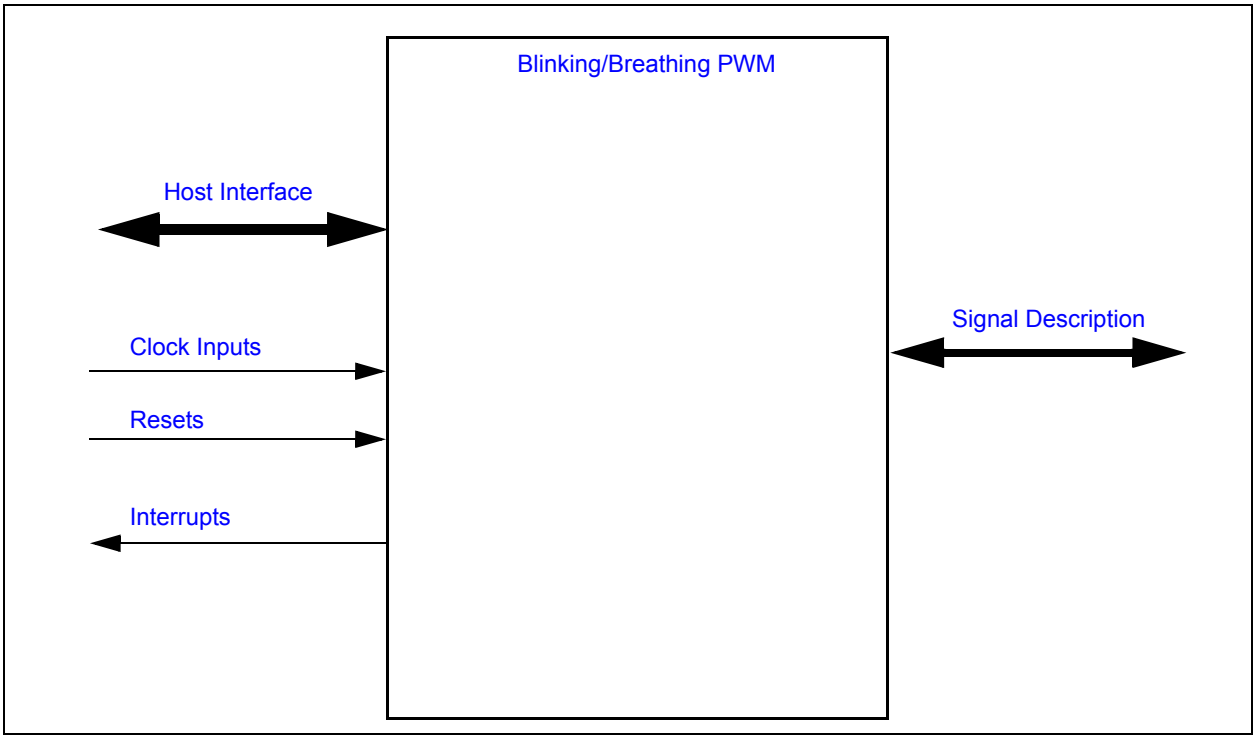
Features:

- 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

21.2 Interface

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

FIGURE 21-1: I/O DIAGRAM OF BLOCK



22.10.2.2 Asserting PREDRIVE_ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Enable Predrive function (PREDRIVE_ENABLE = '1')
3. Program buffer type for all KSO pins to "push-pull”
4. Enable Keyscan Interface (KSEN ='0')

22.10.2.3 De-asserting PREDRIVE_ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Program buffer type for all KSO pins to "open-drain”
3. Disable Predrive function (PREDRIVE_ENABLE = '0')
4. Enable Keyscan Interface (KSEN ='0')

22.10.3 INTERRUPT GENERATION

To support interrupt-based processing, an interrupt can optionally be generated on the high-to-low transition on any of the KSI inputs. A running clock is not required to generate interrupts.

22.10.3.1 Runtime interrupt

KSC_INT is the block’s runtime active-high level interrupt. It is connected to the interrupt interface of the Interrupt Aggregator, which then relays interrupts to the EC.

Associated with each KSI input is a status register bit and an interrupt enable register bit. A status bit is set when the associated KSI input goes from high to low. If the interrupt enable bit for that input is set, an interrupt is generated. An Interrupt is de-asserted when the status bit and/or interrupt enable bit is clear. A status bit cleared when written to a ‘1’.

Interrupts from individual KSIs are logically ORed together to drive the KSC_INT output port. Once asserted, an interrupt is not asserted again until either all KSI[7:0] inputs have returned high or the has changed.

22.10.3.2 Wake-up Interrupt

KSC_INT_WAKE is the block’s wakeup interrupt. It is routed to the Interrupt Aggregator.

During sleep mode, i.e., when the bus clock is stopped, a high-to-low transition on any KSI whose interrupt enable bit is set causes the KSC_INT_WAKE to be asserted. Also set is the associated status bit in the EC Clock Required 2 Status Register (EC_CLK_REQ2_STS). KSC_WAKEUP_INT remains active until the bus clock is started.

The aforementioned transition on KSI also sets the corresponding status bit in the KSI STATUS Register. If enabled, a runtime interrupt is also asserted on KSC_INT when the bus clock resumes running.

22.10.4 WAKE PROGRAMMING

Using the Keyboard Scan Interface to ‘wake’ the CEC1302 can be accomplished using either the Keyboard Scan Interface wake interrupt, or using the wake capabilities of the GPIO Interface pins that are multiplexed with the Keyboard Scan Interface pins. Enabling the Keyboard Scan Interface wake interrupt requires only a single interrupt enable access and is recommended over using the GPIO Interface for this purpose.

22.11 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Keyboard Scan Interface. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Base Address Table.

TABLE 22-6: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
Keyboard Scan Interface	0	EC	32-bit internal address space	4000_9C00h

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

TABLE 22-7: EC-ONLY REGISTER SUMMARY

Offset	Register Name
0h	Reserved
4h	KSO Select Register
8h	KSI INPUT Register
Ch	KSI STATUS Register
10h	KSI INTERRUPT ENABLE Register
14h	Keyscan Extended Control Register

22.11.1 KSO SELECT REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
7	KSO_INVERT This bit controls the output level of KSO pins when selected. 0= KSO[x] driven low when selected 1= KSO[x] driven high when selected.	R/W	0b	VCC1_RESET
6	KSEN This field enables and disables keyboard scan 0= Keyboard scan enabled 1= Keyboard scan disabled. All KSO output buffers disabled.	R/W	1h	VCC1_RESET
5	KSO_ALL 0=When key scan is enabled, KSO output controlled by the KSO_SELECT field. 1=KSO[x] driven high when selected.	R/W	0b	VCC1_RESET
4:0	KSO_SELECT This field selects a KSO line (00000b = KSO[0] etc.) for output according to the value off KSO_INVERT in this register. See Table 22-8, "KSO Select Decode"	R/W	0h	VCC1_RESET

TABLE 22-8: KSO SELECT DECODE

KSO Select [4:0]	KSO Selected
00h	KSO00
01h	KSO01
02h	KSO02
03h	KSO03
04h	KSO04
05h	KSO05
06h	KSO06
07h	KSO07
08h	KSO08
09h	KSO09
0Ah	KSO10

26.7 Power, Clocks and Reset

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

26.7.1 POWER DOMAINS

TABLE 26-1: POWER SOURCES

Name	Description
VCC1	The main power well used when the VBAT RAM is accessed by the EC.
VBAT	The power well used to retain memory state while the main power rail is unpowered.

26.7.2 CLOCK INPUTS

No special clocks are required for this block.

26.7.3 RESETS

TABLE 26-2: RESET SIGNALS

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

26.8 Interrupts

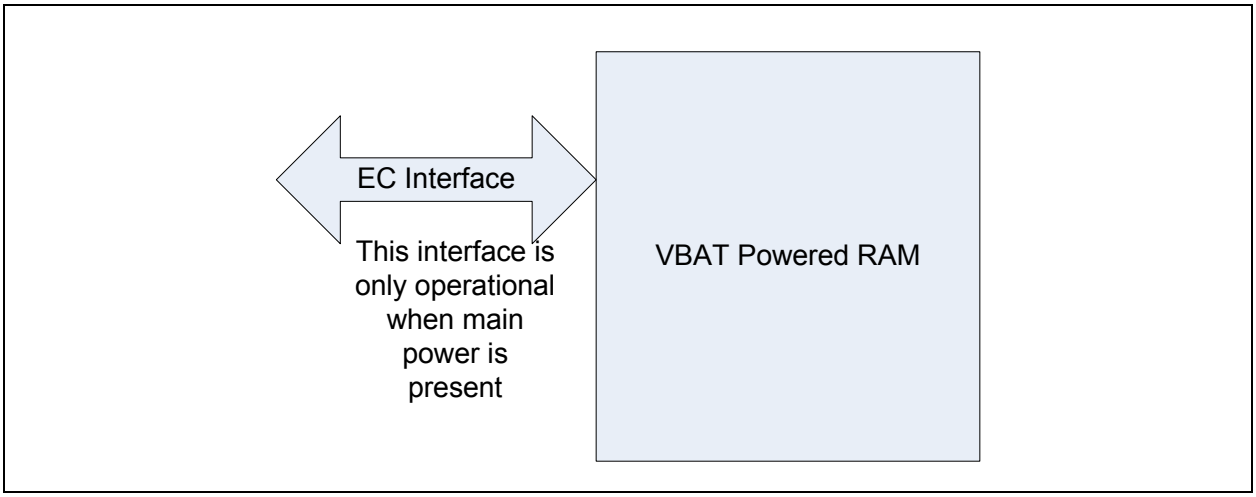
This block does not generate any interrupts.

26.9 Low Power Modes

The VBAT-Powered RAM automatically enters a low power mode whenever it is not being accessed by the EC. There is no chip-level Sleep Enable input.

26.10 Description

FIGURE 26-2: VBAT RAM BLOCK DIAGRAM



The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while VCC1 is powered, and will retain its values powered by VBAT while VCC1 is unpowered. The RAM is organized as a 16 words x 32-bit wide for a total of 64 bytes.

28.2.3 TEST PROCEDURE

28.2.3.1 Setup

Warning: Ensure power supply is off during Setup.

1. Connect JTAG_RST# to ground.
2. Connect the VSS, AVSS, VSS_VBAT pins to ground.
3. Connect the VCC1, AVCC, VBAT pins to an unpowered 3.3V power source.
4. Connect an oscilloscope or voltmeter to the Test Output pin.
5. All other pins should be tied to ground.

Note:	There are 101 pins in the XNOR Chain.

28.2.3.2 Testing

1. Turn on the 3.3V power source.
2. Enable the XNOR Chain as defined in Section 28.2.3.3, "Procedure to Enable the XNOR Chain".

Note:	At this point all inputs to the XNOR Chain are low, except for the JTAG_RST# pin, and the output on the Test Output pin is non-inverted from its initial state, which is dependent on the number of pins in the chain. If the number of input pins in the chain is an even number, the initial state of the Test Output Pin, pin 17: KSO04/GPIO103/TFDP_DATA/XNOR is low. If the number of input pins in the chain is an odd number, the initial state of the Test Output Pin, pin 17: KSO04/GPIO103/TFDP_DATA/XNOR is high.

3. Bring one pin in the chain high. The output on the Test Output Pin, pin 17: KSO04/GPIO103/TFDP_DATA/XNOR pin should toggle. Then individually toggle each of the remaining pins in the chain. Each time an input pin is toggled either high or low the Test Output Pin, pin 17: KSO04/GPIO103/TFDP_DATA/XNOR pin should toggle.
4. Once the XNOR test is completed, exit the XNOR Chain Test Mode by cycling VCC1 power.

28.2.3.3 Procedure to Enable the XNOR Chain

//BEGIN PROCEDURE TO ENTER XNOR CHAIN

////////////////////////////////////

//Reset Test Interface

////////////////////////////////////

force JTAG_RST# = 0

force KSO00/GPIO000/JTAG_TCK = 0

force KSO02/GPIO101/JTAG_TDI = 0

force KSO01/GPIO100/JTAG_TMS = 1

Wait 100 ns

////////////////////////////////

//Come out of reset

////////////////////////////////

force TRST#/JTAG_RST# = 1

Wait 100 ns

force KSO00/GPIO000/JTAG_TCK = 1

force KSO00/GPIO000/JTAG_TCK = 0

force KSO00/GPIO000/JTAG_TCK = 1

force KSO00/GPIO000/JTAG_TCK = 0

29.2.4 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 29-4: 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	µA	
Pull-down current	I_{PD}	39	65	105	µA	
I Type Input Buffer						TTL Compatible Schmitt Trigger Input
Low Input Level	V_{ILI}			0.3x VCC1	V	
High Input Level	V_{IHI}	0.7x VCC1			V	
Tolerance				3.63	V	This buffer is not 5V tolerant.
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}	VCC1-0.4			V	I_{OH} = -2 mA
Tolerance						This buffer is not 5V tolerant.
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	V_{OL} = 2 mA
Tolerance						This buffer is not 5V tolerant.
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
High Output Level	V_{OH}	VCC1-0.4			V	I_{OH} = -4 mA
Tolerance						This buffer is not 5V tolerant.
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	V_{OL} = 4 mA
Tolerance						This buffer is not 5V tolerant.
IOD-4 mA Type Buffer	–	–	–	–	–	Same characteristics as an I and an OD-4mA.

30.14 JTAG Interface Timing

FIGURE 30-22: JTAG POWER-UP & ASYNCHRONOUS RESET TIMING

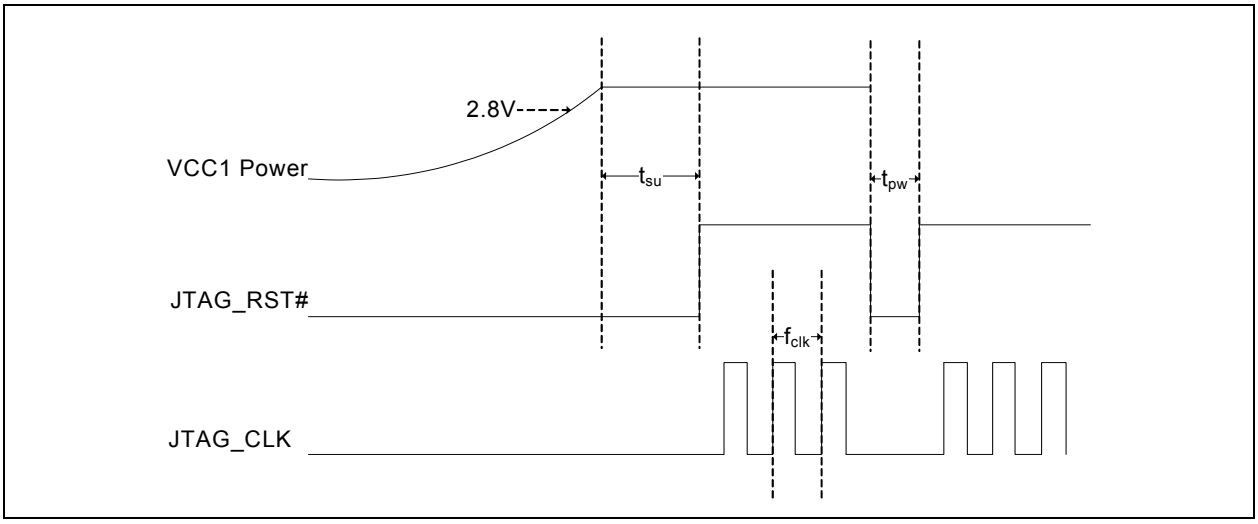


FIGURE 30-23: JTAG SETUP & HOLD PARAMETERS

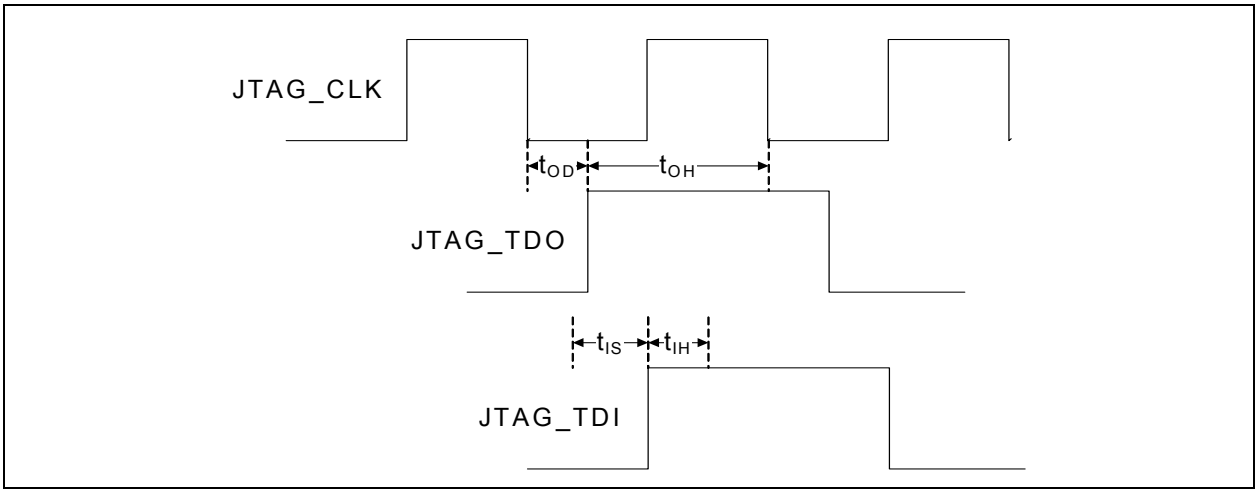


TABLE 30-17: JTAG INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{su}	JTAG_RST# de-assertion after VCC1 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.