MEC1609/MEC1609i

Mixed Signal Mobile Embedded Flash ARC EC BC-Link/ VLPC Base Component

Product Features

- 3.3V Operation
- ACPI Compliant
- LPC Interface
  - LPC I/O and Trusted Cycles Decoded
- VTR (standby) and VBAT (Power Planes)
  - Low Standby Current in Sleep Mode
- Configuration Register Set
  - Compatible with ISA Plug-and-Play Standard
  - EC-Programmable Base Address
- ARC-625D Embedded Controller (EC)
  - 16 KB Single Cycle 32-bit Wide Dual-ported SRAM, Accessible as Closely Coupled Data Memory and Instruction Memory
  - 2 KB Instruction Cache and AHB Memory-mapped SPI Flash Read Controller
  - 32 x 32 x 64 Fast Multiply
  - Divide Assist and Saturation Arithmetic
  - Maskable Interrupt Aggregator/Accelerator Interface
  - Maskable Hardware Wake-Up Events
  - Sleep mode
  - JTAG Debug Port, Includes JTAG Master
  - MCU Serial Debug Port
  - 8-Channel DMA Interface Supports SMBus Controllers and EC/Host GP-SPI Controllers
- Embedded Flash
  - 192 KB user space + 2kB info block, 32-bit Access, 35ns Access Time, 1 K Cycles Endurance
  - Programmable by LPC, EC and JTAG Interfaces
  - Flash Security Enhancements
    - 4K Boot Block Protection
    - Direct JTAG and Direct LPC-protected (2) Pages at or Near Top of Memory for Password Protection
- Legacy Support
  - Fast GATEA20 & Fast CPU_RESET
- System to EC Message Interface
  - 8042 Style Host Interface
  - Embedded Memory Interface
    - Host Serial or Parallel IRQ Source
    - Provides Two Windows to On-Chip SRAM for Host Access
    - Two Register Mailbox Command Interface
    - Host Access of Virtual Registers Without EC Intervention
  - Mailbox Registers Interface
    - Thirty-two 8-Bit Scratch Registers
    - Two Register Mailbox Command Interface
    - Two Register SMI Source Interface
  - ACPI Embedded Controller Interface
    - Four Instances
    - 1 or 4 Byte Data transfer capable
  - ACPI Power Management Interface
    - SCI Event-Generating Functions
- Battery Backed Resources
  - Power-Fail Status Register
  - 32 KHz Clock Generator
  - Week Alarm Timer Interface with Programmable Wake-up from 1ms to 45 Days
  - VBAT-Powered Control Interface
  - VBAT-Backed 64 Byte Memory
- Three EC-based SMBus 2.0 Host Controllers
  - Allows Master or Dual Slave Operation
  - Controllers are Fully Operational on Standby Power
  - DMA-driven I^2C Network Layer Hardware
  - I^2C Datalink Compatibility Mode
  - Multi-Master Capable
  - Supports Clock Stretching
  - Programmable Bus Speeds
  - 400 KHz Capable
  - Hardware Bus Access “Fairness” Interface
  - SMBus Time-outs Interface
  - 8 x 3 x 3 Port Multiplexing
- PECI Interface 2.0
- 18 x 8 Interrupt Capable Multiplexed Keyboard Scan Matrix
- Three independent Hardware Driven PS/2 Ports
  - Fully functional on Main and/or Suspend Power
  - PS/2 Edge Wake Capable
- 115 General Purpose I/O Pins
  - 8 GPIO Pass-Through Port (GPTP)

- 3-pin LED Interface
  - Programmable Blink Rates
  - Breathing LED Output
  - Operational in EC Sleep State
- Programmable 16-bit Counter/Timer Interface
  - Four Wake-capable 16-bit Auto-reloading Counter/Timer Instances
  - Four Operating Modes per Instance: Timer, One-shot, Event and Measurement.
  - 4 External Inputs, 4 External Outputs
- Hibernation Timer Interface
  - Two 32.768 KHz Driven Timers
  - Programmable Wake-up from 0.5ms to 128 Minutes
- System Watch Dog Timer (WDT)
- Input Capture and Compare Timer
  - 32-bit Free-running timer
  - Six 32-bit Capture Registers
  - Two 32-bit Compare Registers
  - Capture, Compare and Overflow Interrupts
- Microchip's Multipoint VLPC Serial Interconnect Bus Master
  - Forwards LPC transactions to VLPC peripherals
  - Forwards ARC transactions to VLPC peripherals
- BC-Link Interconnection Bus
  - Three High Speed and one Low Speed Bus Masters Controllers
- Two General Purpose Serial Peripheral Interface Controllers (ECGP-SPI)
  - One 3-pin EC-driven Full Duplex Serial Communication Interface
  - One 4-pin EC/Host-driven Full Duplex Serial Communication Interface to SPI Flash Interface
  - Flexible Clock Rates
  - SPI Burst Capable
- SPI Flash Read Controller
  - 4 MB AHB Memory-Mapped address space
  - Supports 2 KB EC Instruction Cache
- FAN Support
  - 8 Programmable Pulse-Width Modulator Outputs
    - Multiple Clock Rates
    - 16-Bit 'On' & 16-Bit 'Off' Counters
  - Four Fan Tachometer Inputs
  - 6 x 2 Capture/Compare Timer Interface
- ADC Interface
  - 10-bit Conversion in 10μs
  - 16 Channels
  - Integral Non-Linearity of ±0.5 LSB; Differential Non-Linearity of ±0.5 LSB
- Two Pin Debug Port with Standard 16C550A Register Interface
  - Accessible from Host and EC
  - Programmable Input/output Pin Polarity Inversion
  - Programmable Main Power or Standby Power Functionality
  - Standard Baud Rates to 115.2 Kbps, Custom Baud Rates to 2 Mbps
- Resistor/Capacitor Identification Detection (RC_ID)
  - Single Pin Interface to External Inexpensive RC Circuit
  - Replacement for Multiple GPIO's
  - Provides 8 Quantized States on One Pin
- Integrated Standby Power Reset Generator
- Clock Generator
  - 32.768 KHz-input Clock
  - operational on Suspend Power
  - Programmable Clock Power Management Control & Distribution
  - 64.52 MHz ±2% Accuracy
- Packages:
  - 144 Pin LFBGA RoHS Compliant package
  - 144 Pin TFBGA RoHS Compliant package
- Operating Temperature
  - The MEC1609 supports the commercial temperature range of 0^{o} C to $+70^{o}$ C
  - The MEC1609i supports the industrial temperature range of -40^{o} C to $+85^{o}$ C

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

Table of Contents

1.0 General Description 5
2.0 Pin Configuration 8
3.0 Bus Hierarchy 44
4.0 Logical Device Configuration 53
5.0 Power, Clocks and Resets 73
6.0 Host Interface 113
7.0 Embedded Memory Interface 134
8.0 VLPC Bus Interface 153
9.0 ACPI Embedded Controller Interface 162
10.0 8042 Emulated Keyboard Controller 170
11.0 ACPI PM1 Block Interface 185
12.0 MailBox Register Interface 193
13.0 Two Pin Serial Port (UART) 200
14.0 Embedded Flash Subsystem 219
15.0 ARC 625D Embedded Controller 244
16.0 EC Interrupt Aggregator 251
17.0 Watchdog Timer Interface 293
18.0 EC AHB SPI Flash Read Controller 299
19.0 16-Bit Timer Interface 305
20.0 Hibernation Timer 320
21.0 Week Alarm Interface 324
22.0 GPIO Interface 329
23.0 Input Capture and Compare Timer 347
24.0 DMA Controller 360
25.0 SMB Device Interface 372
26.0 PECI Interface 375
27.0 Analog to Digital Converter (ADC) 378
28.0 TACH Monitor 388
29.0 PWM Controller 397
30.0 RC Identification Detection (RC_ID) 403
31.0 General Purpose Serial Peripheral Interface (GP-SPI) 413
32.0 VBAT-Powered Control Interface 437
33.0 VBAT Powered RAM 442
34.0 LED Interface 444
35.0 PS/2 Device Interface 451
36.0 Keyboard Matrix Scan Support 459
37.0 BC-Link Master 465
38.0 Serial Debug Port 473
39.0 JTAG and XNOR 477
40.0 Electrical Specifications 499
41.0 Timing Diagrams 506
42.0 Reference Documents 525
Appendix A: Data sheet Revision History 526

1.0 GENERAL DESCRIPTION

The MEC1609/MEC1609i is the mixed signal base component of a multi-device advanced I/O controller architecture. The MEC1609/MEC1609i incorporates a high-performance 32-bit ARC 625 embedded microcontroller with a 192 Kilobyte embedded Flash subsystem, 16 Kilobytes of SRAM and 2 Kilobytes of instruction cache with an AHB memory-mapped SPI Flash Read Controller. The MEC1609 communicates with the system host using the Intel® Low Pin Count bus.

There are two distinct protocols that provide communication between the MEC1609/MEC1609i base component and companion components: BC-Link and VLPC. BC-Link in the MEC1609/MEC1609i can access up to four companion components. The BC-Link protocol is peer-to-peer providing communication between the MEC1609/MEC1609i embedded controller and registers located in a companion. VLPC is a multi-drop protocol that matches the MEC1609/MEC1609i with up to three untrusted companion components and one trusted companion component. The MEC1609/MEC1609i accepts LPC Host (ICH/PCH) transactions targeting blocks internal to the MEC1609/MEC1609i and blocks physically located in VLPC companions. The ARC 625 embedded microcontroller can also access blocks that are physically located in VLPC companion components.

The MEC1609/MEC1609i is directly powered by two separate suspend supply planes (VBAT and VTR) and senses a third runtime power plane (VCC) to provide “instant on” and system power management functions. The MEC1609/MEC1609i also contains an integrated VTR Reset Interface and a system Power Management Interface that supports low-power states and can drive state changes as a result of hardware wake events as defined by the MEC1609/MEC1609i Wake Interface.

The MEC1609/MEC1609i defines a software development system interface that includes an MCU Serial Debug Port, a two pin serial debug port with a 16C550A register interface that is accessible to the EC or to the LPC host and can operate up to 2 MB/s, a flexible Flash programming interface and a JTAG interface. The EC can also drive the JTAG interface as a master.

A top-level block diagram of the MEC1609/MEC1609i is shown below in Figure 1-1. An example of system level connection is shown in Figure 1-2. A detailed description of the Bus Hierarchy can be found in Section 3.0, "Bus Hierarchy," on page 44.

FIGURE 1-1: MEC1609/MEC1609I TOP-LEVEL BLOCK DIAGRAM

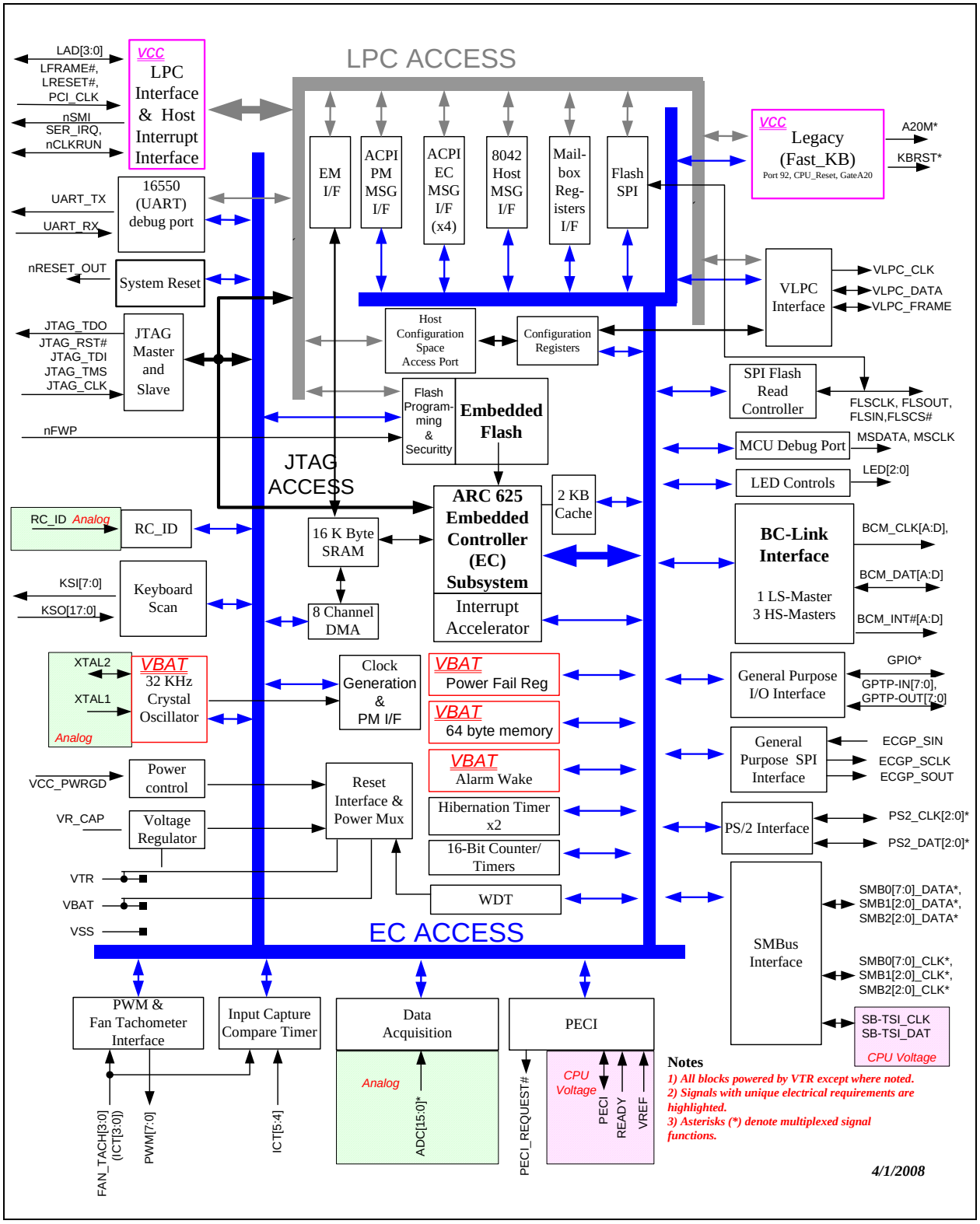
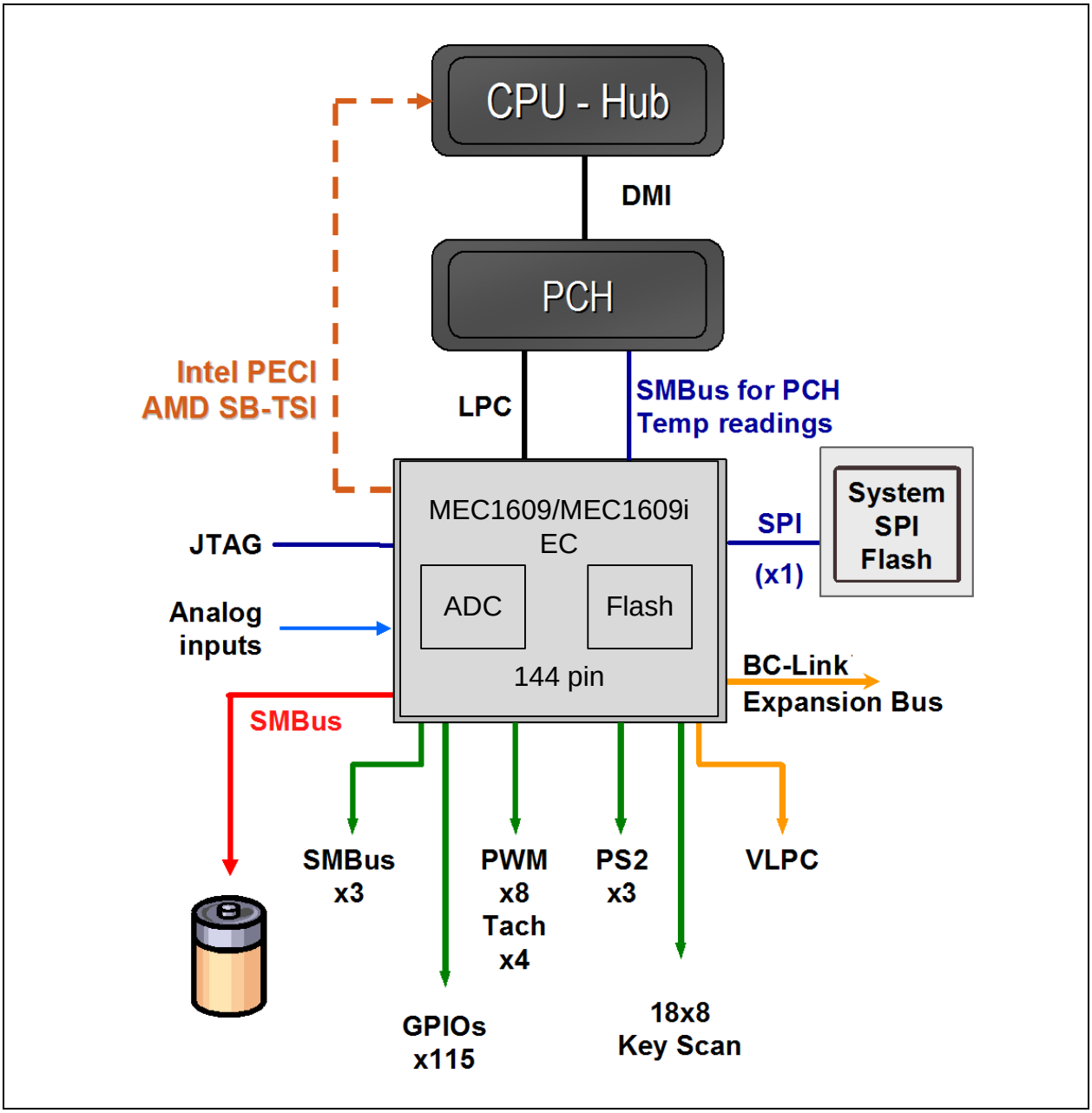


FIGURE 1-2: EXAMPLE OF MEC1609/MEC1609I'S CONNECTIONS TO SYSTEM COMPONENTS



2.0 PIN CONFIGURATION

2.1 Description

The Pin Configuration chapter includes a Pin List, General System/Layout Issues, Pin Description, Pin Multiplexing, Notes for Tables in this Chapter, Strapping Option and Package Outlines.

Note that unless otherwise noted ball numbers shown in the tables throughout this document refer to the 144-Pin LFBGA 10x10x0.8 mm Package Outline (1.4 mm Height).

2.2 Pin List

The MEC1609/MEC1609i Pin List is illustrated below in Table 2-1 and Table 2-2. The LFBGA package ball mapping to MEC1609/MEC1609i Pin Names is shown in Figure 2-1.

TABLE 2-1: PRELIMINARY MEC1609/MEC1609I PIN CONFIGURATION (PIN REF. NUMBERS 1 - 72)

Pin Ref. Number	TFBGA Ball Number	LFBGA Ball Number	Pin Name	Pin Ref. Number	TFBGA Ball Number	LFBGA Ball Number	Pin Name
1	D3	D4	XTAL1	37	N1	K3	ADC11/GPIO213
2	E2	E3	AGND	38	M2	J4	AVTR_ADC
3	D2	D3	XTAL2	39	N2	K4	ADC4/GPIO204
4	C3	C4	VBAT	40	M3	L2	ADC12/GPIO214
5	B1	C1	BGPO0	41	L3	M1	ADC5/GPIO205
6	A2	D1	VCI_OUT	42	N3	M2	ADC13/GPIO215
7	C1	C2	VCI_IN2#	43	M4	L3	ADC6/GPIO206
8	D5	F4	VCI_IN1#	44	L4	L4	ADC14/GPIO216
9	D1	D2	VCI_IN0#	45	N4	M3	ADC7/GPIO207
10	E1	E2	VCI_OVRD_IN	46	N5	M4	ADC15/GPIO217
11	D6	E1	VCI_IN3#	47	K5	J5	VSS_ADC
12	D7	E7	GPIO160/32KHZ_OUT/KSO17	48	L5	L5	LRESET#
13	F2	F3	VCC_PWRGD	49	M5	K5	CLKRUN#
14	F4	F2	GPIO106/nRESET_OUT	50	M6	K6	LFRAME#
15	F1	F7	GPIO101/ECGP_SCLK	51	N6	J6	LDRQ#
16	G4	G7	GPIO102/ECGP_SOUT	52	L6	M5	SER_IRQ
17	F3	G6	GPIO103/ECGP_SIN	53	K6	H7	VTR
18	G2	G3	VSS_RO	54	N7	M6	PCI_CLK
19	G1	F1	VTR	55	M7	L6	LAD0
20	C2	H6	VSS	56	L7	L7	LAD1
21	G3	E8	GPIO021/RC_ID/KSI2	57	N8	K7	LAD2
22	H1	G2	VTR_REG	58	M8	J7	LAD3
23	H2	G1	VR_CAP	59	N9	J8	GPIO100/nEC_SCI
24	H4	G5	GPIO060/KBRST	60	K7	M7	GPIO011/nSMI
25	J2	G4	GPIO127/A20M	61	L8	K8	GPIO061/LPCPD#
26	J1	H4	GPIO116/MSDATA	62	M9	M8	nFWP
27	H3	H5	GPIO117/MSCLK	63	N10	L8	GPIO050/FAN_TACH0
28	K1	H1	AVTR_ADC	64	K9	M9	GPIO051/FAN_TACH1
29	J4	H3	VREF_ADC	65	L9	L9	GPIO052/FAN_TACH2
30	K2	J3	ADC0/GPIO200	66	M10	K9	GPIO016/GPTP-IN7/FAN_TACH3
31	J3	H2	ADC8/GPIO210	67	L10	L10	GPIO053/PWM0
32	L1	J1	ADC1/GPIO201	68	M11	M10	GPIO054/PWM1
33	K4	J2	ADC9/GPIO211	69	N11	M11	GPIO055/PWM2
34	L2	K2	ADC2/GPIO202	70	M12	L11	GPIO056/PWM3
35	M1	K1	ADC10/GPIO212	71	N12	M12	GPIO001/PWM4
36	K3	L1	ADC3/GPIO203	72	L11	K10	GPIO002/PWM5

TABLE 2-2: PRELIMINARY MEC1609/MEC1609I PIN CONFIGURATION (PIN REF. NUMBERS 73 - 144)

Pin Ref. Number	TFBGA Ball Number	LFBGA Ball Number	Pin Nmae	Pin Ref. Number	TFBGA Ball Number	LFBGA Ball Number	Pin Name
73	N13	L12	GPIO014/GPTP-IN6/PWM6	109	F10	A10	GPIO105/UART_RX
74	L12	K12	GPIO015/GPTP-OUT6/PWM7	110	A12	B10	GPIO025/UART_CLK/TIN0/EM_INT
75	K12	F8	GPIO151/GPTP-IN3/ICT4/KSO15	111	B10	B9	GPIO026/GPTP-IN0/TIN1/KSI3
76	M13	E9	GPIO152/GPTP-OUT3/ICT5/KSO16	112	C10	D9	GPIO027/GPTP-OUT0/TIN2/KSI4
77	G10	J9	VTR	113	A11	C9	GPIO030/GPTP-IN1/TIN3/KSI5
78	K10	J10	GPIO003/SMB00_DATA	114	D10	A9	GPIO107/KSO4
79	L13	H10	GPIO004/SMB00_CLK	115	C9	D8	GPIO120/KSO7
80	K13	K11	GPIO005/SMB01_DATA	116	A10	B8	GPIO124/GPTP-OUT4/KSO11
81	K11	J12	GPIO006/SMB01_CLK	117	B9	A8	GPIO125/GPTP-IN4/KSO12
82	J11	J11	GPIO012/SMB07_DATA/SMB22_DATA	118	E10	C8	GPIO031/GPTP-OUT1/TOUT0/KSI6
83	J13	H11	GPIO013/SMB07_CLK/SMB22_CLK	119	C8	B7	GPIO032/GPTP-IN2/TOUT1/KSI7
84	J12	H12	GPIO130/SMB12_DATA	120	B8	D7	GPIO040/GPTP-OUT2/TOUT2/KSO0
85	H12	G10	GPIO131/SMB12_CLK	121	A9	A7	GPIO017/GPTP-OUT7/TOUT3/KSI0
86	J10	G11	GPIO132/SMB06_DATA/KSO14	122	D9	F6	GPIO022/BCM_B_CLK/V_CLK
87	G12	G12	GPIO140/SMB06_CLK	123	A8	E6	GPIO023/BCM_B_DAT/V_DATA
88	H13	H9	VTR_FLASH	124	B7	F5	GPIO024/BCM_B_INT#/V_FRAME
89	H11	G9	GPIO141/SMB05_DATA/SMB20_DATA/FLSCLK	125	D8	A6	GPIO045/LSBCM_D_INT#/KSO1
90	H10	H8	GPIO142/SMB05_CLK/SMB20_CLK/FLSOUT	126	C7	C7	GPIO046/LSBCM_D_DAT/KSO2
91	F12	G8	GPIO143/SMB04_DATA/Reserved/FLSIN	127	A7	B6	GPIO047/LSBCM_D_CLK/KSO3
92	G13	F9	GPIO144/SMB04_CLK/Reserved/FLSCS	128	C6	C6	GPIO121/BCM_A_INT#/KSO8
93	E12	F11	GPIO007/SMB03_DATA/PS2_CLK0B	129	B6	A5	GPIO122/BCM_A_DAT/KSO9
94	F13	F10	GPIO010/SMB03_CLK/PS2_DAT0B	130	E4	E5	VTR
95	F11	F12	GPIO154/SMB02_DATA/PS2_CLK1B	131	A6	B5	GPIO123/BCM_A_CLK/KSO10
96	D12	E12	GPIO155/SMB02_CLK/PS2_DAT1B	132	C5	A4	GPIO041/PECI_REQUEST#
97	G11	E11	GPIO110/PS2_CLK2/GPTP-IN5	133	B5	A3	GPIO042/BCM_C_INT#/PECI_DAT/SB-TSI_DAT
98	E13	E10	GPIO111/PS2_DAT2/GPTP-OUT5	134	A5	A2	GPIO043/BCM_C_DAT/PECI_RDY/SB-TSI_CLK
99	E11	D12	GPIO112/PS2_CLK1A/KSO5	135	A4	A1	GPIO044/BCM_C_CLK/VREF_PECI
100	D13	D11	GPIO113/PS2_DAT1A/KSO6	136	B4	B4	GPIO126/KSO13
101	C12	C12	GPIO114/PS2_CLK0A	137	B3	D6	GPIO020/KSI1
102	C11	D10	GPIO115/PS2_DAT0A	138	B2	B3	GPIO156/LED0
103	C13	C11	GPIO145/SMB11_DATA/JTAG_TDI	139	A3	B2	GPIO157/LED1
104	B12	B12	GPIO146/SMB11_CLK/JTAG_TDO	140	A1	B1	GPIO153/LED2
105	B13	B11	GPIO147/SMB10_DATA/SMB21_DATA/JTAG_CLK	141	C4	D5	VSS
106	D11	A12	GPIO150/SMB10_CLK/SMB21_CLK/JTAG_TMS	142	D4	C3	NO_CONNECT
107	A13	A11	JTAG_RST#	143	E3	C5	NO_CONNECT
108	B11	C10	GPIO104/UART_TX	144	K8	E4	NO_CONNECT

APPLICATION NOTE: in the TFBGA package the “NO_CONNECT” pins (Pin Ref. Numbers 142 - 144) in Table 2-2 must be connected to VSS.

FIGURE 2-1: MEC1609/MEC1609I PIN NAME TO BALL MAPPING (10 MM X 10 MM BGA BOTTOM VIEW)

12	11	10	9	8	7	6	5	4	3	2	1	
GPIO150/SMB10_CLK/SMB21_CLK/JTAG_TMS	JTAG_RST#	GPIO105/UART_RX	GPIO107/KSO4	GPIO125/GPTP-IN4/KSO12	GPIO017/GPTP-OUT7/TOUT3/KSI0	GPIO045/LSBCM_D_INT#/KSO1	GPIO122/BCM_A_DAT/KSO9	GPIO041/PECI_REQUEST#	GPIO042/BCM_C_INT#/PECI_DAT/SB-TSI_DAT	GPIO043/BCM_C_DAT/PECI_RDY/SB-TSI_CLK	GPIO044/BCM_C_CLK/VREF_PECI	**A**
GPIO146/SMB11_CLK/JTAG_TDO	GPIO147/SMB10_DATA/SMB21_DATA/JTAG_CLK	GPIO025/UART_CLK/TIN0/EM_INT	GPIO026/GPTP-IN0/TIN1/KSI3	GPIO124/GPTP-OUT4/KSO11	GPIO032/GPTP-IN2/TOUT1/KSI7	GPIO047/LSBCM_D_CLK/KSO3	GPIO123/BCM_A_CLK/KSO10	GPIO126/KSO13	GPIO156/LED0	GPIO157/LED1	GPIO153/LED2	**B**
GPIO114/PS2_CLK0A	GPIO145/SMB11_DATA/JTAG_TDI	GPIO104/UART_TX	GPIO030/GPTP-IN1/TIN3/KSI5	GPIO031/GPTP-OUT1/TOUT0/KSI6	GPIO046/LSBCM_D_DAT/KSO2	GPIO121/BCM_A_INT#/KSO8	NO_CONNECT	VBAT	NO_CONNECT	VCI_IN2#	BGPO0	**C**
GPIO112/PS2_CLK1A/KSO5	GPIO113/PS2_DAT1A/KSO6	GPIO115/PS2_DAT0A	GPIO027/GPTP-OUT0/TIN2/KSI4	GPIO120/KSO7	GPIO040/GPTP-OUT2/TOUT2/KSO0	GPIO020/KSI1	VSS	XTAL1	XTAL2	VCI_IN0#	VCI_OUT	**D**
GPIO155/SMB02_CLK/PS2_DAT1B	GPIO110/PS2_CLK2/GPTP-IN5	GPIO111/PS2_DAT2/GPTP-OUT5	GPIO152/GPTP-OUT3/ICT5/KSO16	GPIO021/RC_ID/KSI2	GPIO160/32KHZ_OUT/KSO17	GPIO023/BCM_B_DAT/V_DATA	VTR	NO_CONNECT	AGND	VCI_OVRD_IN	VCI_IN3#	**E**
GPIO154/SMB02_DATA/PS2_CLK1B	GPIO007/SMB03_DATA/PS2_CLK0B	GPIO010/SMB03_CLK/PS2_DAT0B	GPIO144/SMB04_CLK/Reserved/FLSCS	GPIO151/GPTP-IN3/ICT4/KSO15	GPIO101/ECGP_SCLK	GPIO022/BCM_B_CLK/V_CLK	GPIO024/BCM_B_INT#/V_FRAME	VCI_IN1#	VCC_PWRGD	GPIO106/nRESET_OUT	VTR	**F**
GPIO140/SMB06_CLK	GPIO132/SMB06_DATA/KSO14	GPIO131/SMB12_CLK	GPIO141/SMB05_DATA/SMB20_DATA/FLSCLK	GPIO143/SMB04_DATA/Reserved/FLSIN	GPIO102/ECGP_SOUT	GPIO103/ECGP_SIN	GPIO060/KBRST	GPIO127/A20M	VSS_RO	VTR_REG	VR_CAP	**G**
GPIO130/SMB12_DATA	GPIO013/SMB07_CLK/SMB22_CLK	GPIO004/SMB00_CLK	VTR_FLASH	GPIO142/SMB05_CLK/SMB20_CLK/FLSOUT	VTR	VSS	GPIO117/MSCLK	GPIO116/MSDATA	VREF_ADC	ADC8/GPIO210	AVTR_ADC	**H**
GPIO006/SMB01_CLK	GPIO012/SMB07_DATA/SMB22_DATA	GPIO003/SMB00_DATA	VTR	GPIO100/nEC_SCI	LAD3	LDRQ#	VSS_ADC	AVTR_ADC	ADC0/GPIO200	ADC9/GPIO211	ADC1/GPIO201	**J**
GPIO015/GPTP-OUT6/PWM7	GPIO005/SMB01_DATA	GPIO002/PWM5	GPIO016/GPTP-IN7/FAN_TACH3	GPIO061/LPCPD#	LAD2	LFRAME#	CLKRUN#	ADC4/GPIO204	ADC11/GPIO213	ADC2/GPIO202	ADC10/GPIO212	**K**
GPIO014/GPTP-IN6/PWM6	GPIO056/PWM3	GPIO053/PWM0	GPIO052/FAN_TACH2	GPIO050/FAN_TACH0	LAD1	LAD0	LRESET#	ADC14/GPIO216	ADC6/GPIO206	ADC12/GPIO214	ADC3/GPIO203	**L**
GPIO001/PWM4	GPIO055/PWM2	GPIO054/PWM1	GPIO051/FAN_TACH1	nFWP	GPIO011/nSMI	PCI_CLK	SER_IRQ	ADC15/GPIO217	ADC7/GPIO207	ADC13/GPIO215	ADC5/GPIO205	**M**

2.3 General System/Layout Issues

2.3.1 PIN DEFAULT STATE THROUGH POWER TRANSITIONS

The power state and power state transitions illustrated in Table 2-3 are defined in Section 5.0, "Power, Clocks and Resets". Pin behavior in this table assumes no specific programming to change the pin state. All GPIO default pins have the same behavior described in Table 2-3 as generically as GPIOXXX.

TABLE 2-3: Pin Default State Through Power Transitions

Pin Reference Number	Signal	VBAT applied	VBAT STABLE	VTR applied	nSYS_RST de-asserted	VCC_PWRGD asserted	VCC_PWRGD de-asserted	nSYS_RST asserted	VTR un-powered	VBAT un-powered	Notes
11	VCI_IN3#	glitch	*In*	*In*	*In*	*In*	*In*	*In*	*In*	glitch	
various	GPIOXXX	unpowered	unpowered	glitch	*In*	*In*	*In*	*Z*	glitch	unpowered	
51	LDRQ#	unpowered	unpowered	glitch	*In*	1>I/O (*P*) >1	*In*	*Z*	glitch	unpowered	Note A
52	SER_IRQ	unpowered	unpowered	glitch	*In*	Z>I/O (*P*) >Z	*In*	*In*	glitch	unpowered	Note A
48	LRESET#	unpowered	unpowered	glitch	*In*	*In*	*In*	Z	glitch	unpowered	Note A
54	PCI_CLK	unpowered	unpowered	glitch	*In*	*In*	*In*	Z	glitch	unpowered	
50	LFRAME#	unpowered	unpowered	glitch	*In*	*In*	*In*	Z	glitch	unpowered	
55	LAD0	unpowered	unpowered	glitch	*In*	In>I/O (*P*) >In	*In*	Z	glitch	unpowered	
56	LAD1	unpowered	unpowered	glitch	*In*	In>I/O (*P*) >In	*In*	Z	glitch	unpowered	
57	LAD2	unpowered	unpowered	glitch	*In*	In>I/O (*P*) >In	*In*	Z	glitch	unpowered	
58	LAD3	unpowered	unpowered	glitch	*In*	In>I/O (*P*) >In	*In*	Z	glitch	unpowered	
49	CLKRUN#	unpowered	unpowered	glitch	*In*	Z>I/O (*P*) >Z	*In*	Z	glitch	unpowered	
5	BGPO0	glitch	Out=0	*Retain*	*Retain*	*Retain*	*Retain*	*Retain*	*Retain*	glitch	*Note B*
7	VCI_IN2#	glitch	*In*	*In*	*In*	*In*	*In*	*In*	*In*	glitch	
6	VCI_OUT	glitch	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	glitch	*Note C*
1	XTAL1	crystal in	crystal in	crystal in	crystal in	crystal in	crystal in	crystal in	crystal in	crystal in	
62	nFWP	glitch	*In*	*In*	*In*	*In*	*In*	*In*	*In*	glitch	Note D
3	XTAL2	crystal out	crystal out	crystal out	crystal out	crystal out	crystal out	crystal out	crystal out	crystal out	
8	VCI_IN1#	glitch	*In*	*In*	*In*	*In*	*In*	*In*	*In*	glitch	
9	VCI_IN0#	glitch	*In*	*In*	*In*	*In*	*In*	*In*	*In*	glitch	
10	VCI_OVRD_IN	glitch	*In*	*In*	*In*	*In*	*In*	*In*	*In*	glitch	

Legend	
(P) = I/O state is driven by protocol while power is	
Z = Tristate	

Notes	
Note A:	This pin exhibits "VCC" power domain emulation
Note B:	This pin is programmable by the EC and will retain its value through a VTR power
Note C:	This pin is prgrammable by the EC and effected by other VBAT inputs pins.
Note D:	This pin exhibits "VTR" power domain emulation

2.3.2 ALTERNATE FUNCTION PIN STATE THROUGH POWER TRANSITIONS

The power state and power state transitions illustrated in Table 2-4 are defined in Section 5.0, "Power, Clocks and Resets". Pin behavior in this table assumes that the EC programs alternate function pin state (see Section 2.5, "Pin Multiplexing," on page 21).

TABLE 2-4: Alternate Function Pin State Through Power Transitions

Pin Reference Number	Signal	VBAT applied	VBAT STABLE	VTR applied	nSYS_RST de-asserted Note E	VCC_PWRGD asserted	VCC_PWRGD de-asserted	nSYS_RST asserted	VTR un-powered	VBAT un-powered	Notes
60	nSMI	unpowered	unpowered	glitch	*In>OD(1)*	1>OD(P)>1	*OD(1)*	*In*	glitch	unpowered	
24	KBRST	unpowered	unpowered	glitch	*In>Z*	1>OD(P)>1	*Z*	*Z>In*	glitch	unpowered	Note F
25	A20M	unpowered	unpowered	glitch	*In>Z*	1>OD(P)>1	*Z*	Z	glitch	unpowered	Note F
61	LPCPD#	unpowered	unpowered	glitch	*In>Z*	In	*Z*	Z	glitch	unpowered	Note F

Legend

(P) = I/O state is driven by protocol while power is

Z = Tristate

OD = Open Drain Output Undriven (1) or driven (0)

Notes

Note E: Transition occurs due to EC selecting alternate function.

Note F: This pin is programmable by the EC and will retain its value through a VTR power

2.3.3 NON 5 VOLT TOLERANT PINS

Table 2-5 lists all signal pins which are not 5.5 Volt tolerant; all other signal pins are 5 Volt tolerant. Signals in Table 2-5 refer to Pin Reference Numbers as defined in Table 2-1 and Table 2-2.

TABLE 2-5: NON 5 VOLT TOLERANT PINS

	LFBGA Ball	Pin Reference Number	Pin Name
1	D4	1	XTAL1
2	D3	3	XTAL2
3	C1	5	BGPO0
4	D1	6	VCI_OUT
5	C2	7	VCI_IN2#
6	F4	8	VCI_IN1#
7	D2	9	VCI_IN0#
8	E2	10	VCI_OVRD_IN
9	E1	11	VCI_IN3#
10	E7	12	GPIO160/32KHZ_OUT/KSO17
11	J3	30	ADC0/GPIO200
12	H2	31	ADC8/GPIO210
13	J1	32	ADC1/GPIO201
14	J2	33	ADC9/GPIO211
15	K2	34	ADC2/GPIO202
16	K1	35	ADC10/GPIO212
17	L1	36	ADC3/GPIO203
18	K3	37	ADC11/GPIO213
19	K4	39	ADC4/GPIO204
20	L2	40	ADC12/GPIO214
21	M1	41	ADC5/GPIO205
22	M2	42	ADC13/GPIO215
23	L3	43	ADC6/GPIO206
24	L4	44	ADC14/GPIO216
25	M3	45	ADC7/GPIO207
26	M4	46	ADC15/GPIO217
27	L5	48	LRESET#
28	K5	49	CLKRUN#
29	K6	50	LFRAME#
30	J6	51	LDRQ#
31	M5	52	SER_IRQ
32	M6	54	PCI_CLK
33	L6	55	LAD0
34	L7	56	LAD1
35	K7	57	LAD2
36	J7	58	LAD3
37	J8	59	GPIO100/nEC_SCI
38	M8	62	nFWP

2.3.4 GLITCH PROTECTED PINS

Table 2-6 lists pins which have POR output glitch protection. POR output glitch protection ensures that these pins will have a steady-state output during VTR POR. Pins without POR output glitch protection may be susceptible to transitory changes as VTR power is applied. Signals in Table 2-6 refer to Pin Reference Numbers as defined in Table 2-1 and Table 2-2.

TABLE 2-6: POR OUTPUT GLITCH PROTECTION

	Pin Reference Numbers	POR Output Glitch Protection
1.	14	GPIO106/nRESET_OUT
2.	85	GPIO131/SMB22_CLK
3.	136	GPIO126/KSO13
4.	5	BGPO0

2.4 Pin Description

2.4.1 OVERVIEW

The following tables describe the signal functions in the MEC1609/MEC1609i pin configuration. See Section 2.6, "Notes for Tables in this Chapter," on page 40 for notes that are referenced in the Pin Description tables.

2.4.2 HOST INTERFACE

TABLE 2-7: HOST INTERFACE

HOST INTERFACE				(14 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
52	M5	SER_IRQ	Serial IRQ	Note 4, Note 5, Note 2, Note 12
51	J6	LDRQ#	LPC Encoded DMA request	Note 4, Note 5, Note 2, Note 12
48	L5	LRESET#	LPC Reset. LRESET# is the same as the system PCI reset, PCIRST#	Note 5, Note 2, Note 12
54	M6	PCI_CLK	PCI Clock	Note 5, Note 2, Note 12
50	K6	LFRAME#	Frame signal. Indicates start of new cycle and termination of broken cycle	Note 5, Note 3, Note 12
55	L6	LAD0	LPC Multiplexed command, address and data bus Bit 0.	Note 4, Note 5, Note 2
56	L7	LAD1	LPC Multiplexed command, address and data bus Bit 1.	Note 4, Note 5, Note 3
57	K7	LAD2	LPC Multiplexed command, address and data bus Bit 2.	Note 4, Note 5, Note 4
58	J7	LAD3	LPC Multiplexed command, address and data bus Bit 3.	Note 4, Note 5, Note 5
49	K5	CLKRUN#	PCI Clock Control	Note 5, Note 2
59	J8	nEC_SCI	Power Management Event	Note 3, Note 5
61	K8	LPCPD#	The LPC Bus Powerdown Signal.	Note 5
60	M7	nSMI	SMI Output	Note 2
110	B10	EM_INT	EM Interface Interrupt Output	

2.4.3 BC-LINK INTERFACE

TABLE 2-8: BC-LINK INTERFACE

BC-Link				(12 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
131	B5	BCM_A_CLK	BC-Link Master clock	
129	A5	BCM_A_DAT	BC-Link Master data I/O	Note 9
128	C6	BCM_A_INT#	BC-Link Master interrupt	
122	F6	BCM_B_CLK	BC-Link Master clock	
123	E6	BCM_B_DAT	BC-Link Master data I/O	Note 9
124	F5	BCM_B_INT#	BC-Link Master interrupt	
135	A1	BCM_C_CLK	BC-Link Master clock	
134	A2	BCM_C_DAT	BC-Link Master data I/O	Note 9
133	A3	BCM_C_INT#	BC-Link Master interrupt	
125	A6	LSBCM_D_INT#	BC-Link Master clock	
126	C7	LSBCM_D_DAT	BC-Link Master data I/O	Note 9
127	B6	LSBCM_D_CLK	BC-Link Master interrupt	

Note 2-1 For ribbon cable applications, the Low Speed BC-Link Master maximum clock frequency is 3 MHz. The High Speed BC-Link Master maximum clock frequency is 21.5 MHz. The clock frequency is set with the BC Clock Select. register.

Note 2-2 the BCM DAT pins require a weak pull up resistor (100 K Ohms).

2.4.4 JTAG INTERFACE

TABLE 2-9: JTAG INTERFACE

JTAG Interface				(5 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
105	B11	JTAG_CLK	JTAG Test Clock	
107	A11	JTAG_RST#	JTAG Test Reset (active low)	Note 14
103	C11	JTAG_TDI	JTAG Test Data In	
104	B12	JTAG_TDO	JTAG Test Data Out	
106	A12	JTAG_TMS	JTAG Test Mode Select	

2.4.5 MASTER CLOCK INTERFACE

TABLE 2-10: MASTER CLOCK INTERFACE

Master Clock Interface				(3 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
1	D4	XTAL1	32.768 KHz Crystal Input	Note 5
3	D3	XTAL2	32.768 KHz Crystal Output (Single-ended 32.768 KHz Clock Input)	Note 5
12	E7	32KHZ_OUT	32.768 KHz Digital Output	

APPLICATION NOTE: The MEC1609/MEC1609i crystal oscillator design requires a 32.768 KHz parallel resonant 12.5 pF load capacitance crystal with two 22 pF load caps. Refer to Application Note 19.3 PCB Layout Guide for MEC1609/MEC1609i for more information.

APPLICATION NOTE: The MEC1609/MEC1609i does not have a stringent accuracy requirement for the 32K crystal; however, the system may. The accuracy of the 32K input translates directly into accuracy of the internal 32K clock and the functions that use it; e.g., the 32KHZ_OUT, week timer, hibernation timers, etc.

Total accuracy error is based on multiple system design factors including stray capacitance, crystal accuracy etc. The accuracy, with regard to actual error in time can be illustrated as such: ±1ppm of error in frequency corresponds to 32.768 KHz ´ 1ppm x $1´10^{-6}$ = ±0.032768 Hz. This translates into ~1 µsec/sec or ~ ±0.086 sec/day. Choose the crystal accuracy accordingly.

2.4.6 ANALOG DATA ACQUISITION INTERFACE

TABLE 2-11: ANALOG DATA ACQUISITION

Analog Data Acquisition Interface				(17 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
30	J3	ADC0	ADC channel 0	
32	J1	ADC1	ADC channel 1	
34	K2	ADC2	ADC channel 2	
36	L1	ADC3	ADC channel 3	
39	K4	ADC4	ADC channel 4	
41	M1	ADC5	ADC channel 5	
43	L3	ADC6	ADC channel 6	
45	M3	ADC7	ADC channel 7	
31	H2	ADC8	ADC channel 8	
33	J2	ADC9	ADC channel 9	
35	K1	ADC10	ADC channel 10	
37	K3	ADC11	ADC channel 11	
40	L2	ADC12	ADC channel 12	
42	M2	ADC13	ADC channel 13	
44	L4	ADC14	ADC channel 14	
46	M4	ADC15	ADC channel 15	
29	H3	VREF_ADC	ADC Voltage Reference Pin	

Note: The voltage on the pins in Table 2-11 must not exceed 3.6 V or damage to the device will occur.

2.4.7 FAN TACHOMETER, PWM AND INPUT CAPTURE TIMER INTERFACE

TABLE 2-12: FAN PWM & TACHOMETER INTERFACE

FAN PWM & TACHOMETER				(14 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
63	L8	FAN_TACH0	Fan Tachometer Input 0 (Input Capture Timer Input 0)	
64	M9	FAN_TACH1	Fan Tachometer Input 1 (Input Capture Timer Input 1)	
65	L9	FAN_TACH2	Fan Tachometer Input 2 (Input Capture Timer Input 2)	
66	K9	FAN_TACH3	Fan Tachometer Input 3 (Input Capture Timer Input 3)	
75	F8	ICT4	Input Capture Timer Input 4	
76	E9	ICT5	Input Capture Timer Input 5	
67	L10	PWM0	Pulse Width Modulator Output 0	
68	M10	PWM1	Pulse Width Modulator Output 1	
69	M11	PWM2	Pulse Width Modulator Output 2	
70	L11	PWM3	Pulse Width Modulator Output 3	
71	M12	PWM4	Pulse Width Modulator Output 4	
72	K10	PWM5	Pulse Width Modulator Output 5	
73	L12	PWM6	Pulse Width Modulator Output 6	
74	K12	PWM7	Pulse Width Modulator Output 7	

2.4.8 GENERAL PURPOSE I/O INTERFACE

TABLE 2-13: GPIO INTERFACE

General Purpose Input/Output				(115 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
(7), (8), (9), (10), (11), 12, (13), 14, 15, 16, 17, 21, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 45, 46, (48), (49), (50), (51), (52), (54), 59, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140	(C2), (F4), (D2), (E2), (E1), E7, (F3), F2, F7, G7, G6, E8, G5, G4, H4, H5, J3, H2, J1, J2, K2, K1, L1, K3, K4, L2, M1, M2, L3, L4, M3, M4, (L5), (K5), (K6), (J6), (M5), (M6), J8, M7, K8, L8, M9, L9, K9, L10, M10, M11, L11, M12, K10, L12, K12, F8, E9, J10, H10, K11, J12, J11, H11, H12, G10, G11, G12, G9, H8, G8, F9, F11, F10, F12, E12, E11, E10, D12, D11, C12, D10, C11, B12, B11, A12, C10, A10, B10, B9, D9, C9, A9, D8, B8, A8, C8, B7, D7, A7, F6, E6, F5, A6, C7, B6, C6, A5, B5, A4, A3, A2, A1, B4, D6, B3, B2, B1	GPIO	General Purpose Input Output Pins (pin numbers in parentheses represent interrupt-only or non-functional GPIOs)	Note 8, Note 11

2.4.9 GENERAL PURPOSE PASS-THROUGH PORTS INTERFACE

TABLE 2-14: GPIO PASS-THROUGH PORTS

General Purpose Pass-Through Ports				(16 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
111	B9	GPTP-IN0	General Purpose Pass Through Port Input 0	Note 8
113	C9	GPTP-IN1	General Purpose Pass Through Port Input 1	Note 8
119	B7	GPTP-IN2	General Purpose Pass Through Port Input 2	Note 8
75	F8	GPTP-IN3	General Purpose Pass Through Port Input 3	Note 8
117	A8	GPTP-IN4	General Purpose Pass Through Port Input 4	Note 8
97	E11	GPTP-IN5	General Purpose Pass Through Port Input 5	Note 8
73	L12	GPTP-IN6	General Purpose Pass Through Port Input 6	Note 8
66	K9	GPTP-IN7	General Purpose Pass Through Port Input 7	Note 8
112	D9	GPTP-OUT0	General Purpose Pass Through Port Output 0	
118	C8	GPTP-OUT1	General Purpose Pass Through Port Output 1	
120	D7	GPTP-OUT2	General Purpose Pass Through Port Output 2	
76	E9	GPTP-OUT3	General Purpose Pass Through Port Output 3	
116	B8	GPTP-OUT4	General Purpose Pass Through Port Output 4	
98	E10	GPTP-OUT5	General Purpose Pass Through Port Output 5	
74	K12	GPTP-OUT6	General Purpose Pass Through Port Output 6	
121	A7	GPTP-OUT7	General Purpose Pass Through Port Output 7	

2.4.10 MISCELLANEOUS FUNCTIONS

TABLE 2-15: MISCELLANEOUS FUNCTIONS

MISC Functions				(14 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
25	G4	A20M	KBD GATEA20 Output	Note 2
24	G5	KBRST	CPU_RESET	Note 2
138	B3	LED0	LED Output 0	
139	B2	LED1	LED Output 1	
140	B1	LED2	LED Output 2	
27	H5	MSCLK	MCHP Proprietary EC debug port	Note 10
26	H4	MSDATA	MCHP Proprietary EC debug port	Note 10
110	B10	UART_CLK	UART CLK input	
109	A10	UART_RX	UART RX Input	Note 13
108	C10	UART_TX	UART TX Output	Note 14
13	F3	VCC_PWRGD	System Main Power Indication	Note 12
21	E8	RC_ID	RC Identification Detection	
62	M8	nFWP	(Boot) Flash Write Protect	
14	F2	nRESET_OUT	EC-driven External System Reset	

Note 2-3 The KBRST pin function is the output of CPU_RESET described in Section 10.13, "CPU_RESET Hardware Speed-Up," on page 183.

Note 2-4 When the CLK_SRC bit is '1' in the Configuration Select register (pg. 217), the baud clock is externally sourced from the UART_CLK pin. UART_CLK requires a frequency of 1.8432 MHz ± 2%.

Note 2-5 The nRESET_OUT pin function is an external output signal version of the internal signal nSIO_RESET. See the iRESET OUT bit in the PCR Status and Control Register on page 104.

2.4.11 PS/2 INTERFACE

TABLE 2-16: PS/2 INTERFACE

PS/2 Interface				(10 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
93	F11	PS2_CLK0B	PS/2 clock ouput	Note 16
94	F10	PS2_DAT0B	PS/2 data	Note 16
97	E11	PS2_CLK2	PS/2 clock ouput	
98	E10	PS2_DAT2	PS/2 data	
99	D12	PS2_CLK1A	PS/2 clock ouput	Note 16
100	D11	PS2_DAT1A	PS/2 data	Note 16
101	C12	PS2_CLK0A	PS/2 clock ouput	Note 16
102	D10	PS2_DAT0A	PS/2 data	Note 16
95	F12	PS2_CLK1B	PS/2 clock ouput	Note 16
96	E12	PS2_DAT1B	PS/2 data	Note 16

2.4.12 POWER INTERFACE

TABLE 2-17: POWER INTERFACE

Power Interface				(15 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
2	E3	AGND	VBAT associated ground	
4	C4	VBAT	VBAT supply	
23	G1	VR_CAP	Internal Voltage Regulator Output (Capacitor Required)	
20, 141	H6, D5	VSS	VTR associated ground	
18	G3	VSS_RO	VTR associated ground used for ring ocsillator.	
19, 53, 77, 130	F1, H7, J9, E5	VTR	VTR supply	
28, 38	H1, J4	AVTR_ADC	Analog VTR Supply	
47	J5	VSS_ADC	Analog VTR associated ground	
22	G2	VTR_REG	VTR Internal Voltage Regulator Supply	
88	H9	VTR_FLASH	VTR Internal Flash Supply	

APPLICATION NOTE: VBAT to VTR switching must be done externally as described in Section 5.6.7, "Power Mux," on page 98.

2.4.13 SMBUS INTERFACE

TABLE 2-18: SMBUS INTERFACE

SMBus Interface				(30 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
78	J10	SMB00_DATA	SMBus Controller 0 Port 0 Data	
79	H10	SMB00_CLK	SMBus Controller 0 Port 0 Clock	
80	K11	SMB01_DATA	SMBus Controller 0 Port 1 Data	
81	J12	SMB01_CLK	SMBus Controller 0 Port 1 Clock	
95	F12	SMB02_DATA	SMBus Controller 0 Port 2 Data	
96	E12	SMB02_CLK	SMBus Controller 0 Port 2 Clock	
93	F11	SMB03_DATA	SMBus Controller 0 Port 3 Data	
94	F10	SMB03_CLK	SMBus Controller 0 Port 3 Clock	
91	G8	SMB04_DATA	SMBus Controller 0 Port 4 Data	
92	F9	SMB04_CLK	SMBus Controller 0 Port 4 Clock	
89	G9	SMB05_DATA	SMBus Controller 0 Port 5 Data	
90	H8	SMB05_CLK	SMBus Controller 0 Port 5 Clock	
86	G11	SMB06_DATA	SMBus Controller 0 Port 6 Data	
87	G12	SMB06_CLK	SMBus Controller 0 Port 6 Clock	
82	J11	SMB07_DATA	SMBus Controller 0 Port 7 Data	
83	H11	SMB07_CLK	SMBus Controller 0 Port 7 Clock	
105	B11	SMB10_DATA	SMBus Controller 1 Port 0 Data	
106	A12	SMB10_CLK	SMBus Controller 1 Port 0 Clock	
103	C11	SMB11_DATA	SMBus Controller 1 Port 1 Data	
104	B12	SMB11_CLK	SMBus Controller 1 Port 1 Clock	
84	H12	SMB12_DATA	SMBus Controller 1 Port 2 Data	
85	G10	SMB12_CLK	SMBus Controller 1 Port 2 Clock	
89	G9	SMB20_DATA	SMBus Controller 2 Port 0 Data	
90	H8	SMB20_CLK	SMBus Controller 2 Port 0 Clock	
105	B11	SMB21_DATA	SMBus Controller 2 Port 1 Data	
106	A12	SMB21_CLK	SMBus Controller 2 Port 1 Clock	
82	J11	SMB22_DATA	SMBus Controller 2 Port 2 Data	
83	H11	SMB22_CLK	SMBus Controller 2 Port 2 Clock	
133	A3	SB-TSI_DAT	SMBus Controller 2 Port 3 Data	
134	A2	SB-TSI_CLK	SMBus Controller 2 Port 3 Clock	

2.4.14 VLPC INTERFACE

TABLE 2-19: VLPC INTERFACE

VLPC Interface				(3 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
122	F6	V_CLK	VLPC Master Clock	
123	E6	V_DATA	VLPC Data	
124	F5	V_FRAME	VLPC Frame	

2.4.15 PECI INTERFACE

TABLE 2-20: PECI INTERFACE

PECI Interface				(4 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
133	A3	PECI_DAT	PECI Bus	
134	A2	PECI_RDY	PECI Ready Input	
135	A1	VREF_PECI	PECI Interface Voltage Reference	
132	A4	PECI_REQUEST#	PECI Request Output	Note 19

2.4.16 16-BIT COUNTER/TIMER INTERFACE

TABLE 2-21: 16-BIT COUNTER/TIMER INTERFACE

16-Bit Counter/Timer Interface				(8 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
110	B10	TIN0	16-Bit Counter/Timer Input 0	
111	B9	TIN1	16-Bit Counter/Timer Input 1	
112	D9	TIN2	16-Bit Counter/Timer Input 2	
113	C9	TIN3	16-Bit Counter/Timer Input 3	
118	C8	TOUT0	16-Bit Counter/Timer Output 0	
119	B7	TOUT1	16-Bit Counter/Timer Output 1	
120	D7	TOUT2	16-Bit Counter/Timer Output 2	
121	A7	TOUT3	16-Bit Counter/Timer Output 3	

2.4.17 KEYBOARD SCAN INTERFACE

TABLE 2-22: KEYBOARD SCAN INTERFACE

Keyboard Scan Interface				(26 Pins)
Pin Ref. Number	**Ball**	**Signal Name**	**Description**	**Notes**
121	A7	KSI0	Keyboard Scan Matrix Input 0	
137	D6	KSI1	Keyboard Scan Matrix Input 1	
21	E8	KSI2	Keyboard Scan Matrix Input 2	
111	B9	KSI3	Keyboard Scan Matrix Input 3	
112	D9	KSI4	Keyboard Scan Matrix Input 4	
113	C9	KSI5	Keyboard Scan Matrix Input 5	
118	C8	KSI6	Keyboard Scan Matrix Input 6	
119	B7	KSI7	Keyboard Scan Matrix Input 7	
120	D7	KSO0	Keyboard Scan Matrix Output 0	
125	A6	KSO1	Keyboard Scan Matrix Output 1	
126	C7	KSO2	Keyboard Scan Matrix Output 2	
127	B6	KSO3	Keyboard Scan Matrix Output 3	
114	A9	KSO4	Keyboard Scan Matrix Output 4	
99	D12	KSO5	Keyboard Scan Matrix Output 5	
100	D11	KSO6	Keyboard Scan Matrix Output 6	
115	D8	KSO7	Keyboard Scan Matrix Output 7	
128	C6	KSO8	Keyboard Scan Matrix Output 8	
129	A5	KSO9	Keyboard Scan Matrix Output 9	
131	B5	KSO10	Keyboard Scan Matrix Output 10	
116	B8	KSO11	Keyboard Scan Matrix Output 11	
117	A8	KSO12	Keyboard Scan Matrix Output 12	
136	B4	KSO13	Keyboard Scan Matrix Output 13	
86	G11	KSO14	Keyboard Scan Matrix Output 14	
75	F8	KSO15	Keyboard Scan Matrix Output 15	
76	E9	KSO16	Keyboard Scan Matrix Output 16	
12	E7	KSO17	Keyboard Scan Matrix Output 17	

2.4.18 VCI

TABLE 2-23: VCI INTERFACE

VBAT-Powered Control Interface				(7 Pins)
Pin Ref. Number	Ball	Signal Name	Description	Notes
10	E2	VCI_OVRD_IN	Input can cause wakeup or interrupt event	Note 12
6	D1	VCI_OUT	OUTPUT from combinational logic and/or EC	
5	C1	BGPO0	VBAT driven GPO	
9	D2	VCI_IN0#	Input can cause wakeup or interrupt event	Note 12
8	F4	VCI_IN1#	Input can cause wakeup or interrupt event	Note 12
7	C2	VCI_IN2#	Input can cause wakeup or interrupt event	Note 12
11	E1	VCI_IN3#	Input can cause wakeup or interrupt event	Note 12

2.4.19 SPI CONTROLLERS INTERFACE

TABLE 2-24: SPI CONTROLLERS INTERFACE

SPI Controllers Interface				(7 Pins)
Pin Ref. Number	Ball	Signal Name	Description	Notes
15	F7	ECGP_SCLK	General Purpose SPI Clock	
16	G7	ECGP_SOUT	General Purpose SPI Output	
17	G6	ECGP_SIN	General Purpose SPI Input	
89	G9	FLSCLK	Flash Interface SPI Clock	
90	H8	FLSOUT	Flash Interface SPI Output	
91	G8	FLSIN	Flash Interface SPI Input	
92	F9	FLSCS	Flash Interface SPI Chip Select	

Note 2-6 For General Purpose Serial Peripheral Interface (GP-SPI) interface pins with 8 mA buffers the maximum SPCLK pin clock frequency is 16.13 MHz for all modes. Limited functionality is available at 32.26 MHz; although the block can be programmed for higher frequencies, performance is not ensured (see Table 31-14, “SPI_CLK Frequencies,” on page 430 and Section 31.9.5.5, "Limits of SPI configurations," on page 424).

2.5 Pin Multiplexing

Multifunction Pin Multiplexing in the MEC1609/MEC1609i is controlled by the GPIO Interface and illustrated in the Multiplexing Tables [1:18] that follow. See Section 2.6, "Notes for Tables in this Chapter," on page 40 for notes that are referenced in the Pin Multiplexing tables. See Section 22.10.1, "Pin Control Register," on page 337 for Pin Multiplexing programming details. See also Section 22.5, "Pin Multiplexing Control," on page 331.

Pin signal functions that exhibit power domain emulation (see VCC Power Domain Emulation and VTR Power Domain Emulation below) have a different power supply designation in the “Emulated Power Well” column and “Signal Power Well“ columns of the Multiplexing Tables [1:18] in Section 2.5.3. See also Section 2.3.1, "Pin Default State Through Power Transitions," on page 11 for a description of pin states through power transitions.

2.5.1 VCC POWER DOMAIN EMULATION

Pin signal functions that exhibit VCC Power Domain Emulation are documented in the Multiplexing Tables [1:18] as “Signal Power Well“= VTR and “Emulated Power Well” = VCC. The System Runtime Supply power is not connected to the MEC1609/MEC1609i. The VCC_PWRGD signal is used to indicate when power is applied to the System Runtime Supply. All pin signal functions that exhibit VCC power domain emulation are powered by VTR and controlled by the VCC_PWRGD signal input. VCC power domain emulation pin signal functions are tri-stated when VCC_PWRGD is not asserted and are functional when VCC_PWRGD is active.

2.5.2 VTR POWER DOMAIN EMULATION

Pin Signal Functions that exhibit VTR Power Domain Emulation are documented in the Multiplexing Tables [1:18] as “Signal Power Well“= VBAT and “Emulated Power Well” = VTR. All pin signal functions that exhibit VTR Power Domain Emulation are powered by VBAT and controlled by the internal VTR POR. VTR power domain emulation pin signal functions are tristated when VTR power is not applied and are functional when VTR power is applied.

2.5.3 MULTIPLEXING TABLES [1:18]

TABLE 2-25: MULTIPLEXING TABLE (1 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
1	D4	Default: 0	XTAL1	ICLK	VBAT/VTR	VBAT/VTR	
1	D4	1	Reserved	Reserved	Reserved	Reserved	
1	D4	2	Reserved	Reserved	Reserved	Reserved	
1	D4	3	Reserved	Reserved	Reserved	Reserved	
2	E3		AGND	PWR	PWR	PWR	
2	E3						
2	E3						
2	E3						
3	D3	Default: 0	XTAL2	OCLK/ICLK	VBAT/VTR	VBAT/VTR	
3	D3	1	Reserved	Reserved	Reserved	Reserved	
3	D3	2	Reserved	Reserved	Reserved	Reserved	
3	D3	3	Reserved	Reserved	Reserved	Reserved	
4	C4		VBAT	PWR	PWR	PWR	
4	C4						
4	C4						
4	C4						
5	C1	Default: 0	BGPO0	O-8 mA	VBAT	VBAT	
5	C1	1	Reserved	Reserved	Reserved	Reserved	
5	C1	2	Reserved	Reserved	Reserved	Reserved	
5	C1	3	Reserved	Reserved	Reserved	Reserved	
6	D1	Default: 0	VCI_OUT	O	VBAT	VBAT	
6	D1	1	Reserved	Reserved	Reserved	Reserved	
6	D1	2	Reserved	Reserved	Reserved	Reserved	
6	D1	3	Reserved	Reserved	Reserved	Reserved	
7	C2	0	Reserved	Reserved	Reserved	Reserved	
7	C2	Default: 1	VCI_IN2#	I	VBAT	VBAT	Note 12
7	C2	2	Reserved	Reserved	Reserved	Reserved	
7	C2	3	Reserved	Reserved	Reserved	Reserved	
8	F4	0	Reserved	Reserved	Reserved	Reserved	
8	F4	Default: 1	VCI_IN1#	I	VBAT	VBAT	Note 12
8	F4	2	Reserved	Reserved	Reserved	Reserved	
8	F4	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-26: MULTIPLEXING TABLE (2 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
9	D2	0	Reserved	Reserved	Reserved	Reserved	
9	D2	Default: 1	VCI_IN0#	I	VBAT	VBAT	Note 12
9	D2	2	Reserved	Reserved	Reserved	Reserved	
9	D2	3	Reserved	Reserved	Reserved	Reserved	
10	E2	0	Reserved	Reserved	Reserved	Reserved	
10	E2	Default: 1	VCI_OVRD_IN	I	VBAT	VBAT	Note 12
10	E2	2	Reserved	Reserved	Reserved	Reserved	
10	E2	3	Reserved	Reserved	Reserved	Reserved	
11	E1	0	Reserved	Reserved	Reserved	Reserved	
11	E1	Default: 1	VCI_IN3#	I	VBAT	VBAT	Note 12
11	E1	2	Reserved	Reserved	Reserved	Reserved	
11	E1	3	Reserved	Reserved	Reserved	Reserved	
12	E7	Default: 0	GPIO160	(I/O/OD)-8 mA	VTR	VTR	
12	E7	1	32KHZ_OUT	O-8 mA	VTR	VTR	
12	E7	2	KSO17	OD-8 mA	VTR	VTR	
12	E7	3	Reserved	Reserved	Reserved	Reserved	
13	F3	0	Reserved	Reserved	Reserved	Reserved	
13	F3	Default: 1	VCC_PWRGD	I	VTR	VTR	Note 12
13	F3	2	Reserved	Reserved	Reserved	Reserved	
13	F3	3	Reserved	Reserved	Reserved	Reserved	
14	F2	Default: 0	GPIO106	(I/O/OD)-8 mA	VTR	VTR	
14	F2	1	nRESET_OUT	O-8 mA	VTR	VTR	
14	F2	2	Reserved	Reserved	Reserved	Reserved	
14	F2	3	Reserved	Reserved	Reserved	Reserved	
15	F7	Default: 0	GPIO101	(I/O/OD)-8 mA	VTR	VTR	
15	F7	1	ECGP_SCLK	O-8 mA	VTR	VTR	
15	F7	2	Reserved	Reserved	Reserved	Reserved	
15	F7	3	Reserved	Reserved	Reserved	Reserved	
16	G7	Default: 0	GPIO102	(I/O/OD)-8 mA	VTR	VTR	
16	G7	1	ECGP_SOUT	(I/O/OD)-8 mA	VTR	VTR	
16	G7	2	Reserved	Reserved	Reserved	Reserved	
16	G7	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-27: MULTIPLEXING TABLE (3 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
17	G6	Default: 0	GPIO103	(I/O/OD)-8 mA	VTR	VTR	
17	G6	1	ECGP_SIN	(I/O/OD)-8 mA	VTR	VTR	
17	G6	2	Reserved	Reserved	Reserved	Reserved	
17	G6	3	Reserved	Reserved	Reserved	Reserved	
18	G3		VSS_RO	PWR	PWR	PWR	
18	G3						
18	G3						
18	G3						
19	F1		VTR	PWR	PWR	PWR	
19	F1						
19	F1						
19	F1						
20	H6		VSS	PWR	PWR	PWR	
20	H6						
20	H6						
20	H6						
21	E8	Default: 0	GPIO021	(IS/O/OD)-12 mA	VTR	VTR	
21	E8	1	RC_ID	IS	VTR	VTR	
21	E8	2	KSI2	IS	VTR	VTR	
21	E8	3	Reserved	Reserved	Reserved	Reserved	
22	G2		VTR_REG	PWR	PWR	PWR	
22	G2						
22	G2						
22	G2						
23	G1		VR_CAP	PWR	PWR	PWR	
23	G1						
23	G1						
23	G1						
24	G5	Default: 0	GPIO060	(I/O/OD)-8 mA	VTR	VTR	
24	G5	1	KBRST	OD-8 mA	VTR	VCC	
24	G5	2	Reserved	Reserved	Reserved	Reserved	
24	G5	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-28: MULTIPLEXING TABLE (4 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
25	G4	Default: 0	GPIO127	(I/O/OD)-8 mA	VTR	VTR	
25	G4	1	A20M	O-8 mA	VTR	VCC	
25	G4	2	Reserved	Reserved	Reserved	Reserved	
25	G4	3	Reserved	Reserved	Reserved	Reserved	
26	H4	Default: 0	GPIO116	(I/O/OD)-16 mA	VTR	VTR	Note 20
26	H4	1	MSDATA	O-16 mA	VTR	VTR	
26	H4	2	Reserved	Reserved	Reserved	Reserved	
26	H4	3	Reserved	Reserved	Reserved	Reserved	
27	H5	Default: 0	GPIO117	(I/O/OD)-16 mA	VTR	VTR	
27	H5	1	MSCLK	O-16 mA	VTR	VTR	
27	H5	2	Reserved	Reserved	Reserved	Reserved	
27	H5	3	Reserved	Reserved	Reserved	Reserved	
28	H1		AVTR_ADC	PWR	PWR	PWR	
28	H1						
28	H1						
28	H1						
29	H3		VREF_ADC	PWR	PWR	PWR	
29	H3						
29	H3						
29	H3						
30	J3	0	GPIO200	(I/O/OD)-8 mA	VTR	VTR	
30	J3	Default: 1	ADC0	I	VREF	VREF	
30	J3	2	Reserved	Reserved	Reserved	Reserved	
30	J3	3	Reserved	Reserved	Reserved	Reserved	
31	H2	0	GPIO210	(I/O/OD)-8 mA	VTR	VTR	
31	H2	Default: 1	ADC8	I	VREF	VREF	
31	H2	2	Reserved	Reserved	Reserved	Reserved	
31	H2	3	Reserved	Reserved	Reserved	Reserved	
32	J1	0	GPIO201	(I/O/OD)-8 mA	VTR	VTR	
32	J1	Default: 1	ADC1	I	VREF	VREF	
32	J1	2	Reserved	Reserved	Reserved	Reserved	
32	J1	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-29: MULTIPLEXING TABLE (5 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
33	J2	0	GPIO211	(I/O/OD)-8 mA	VTR	VTR	
33	J2	Default: 1	ADC9	I	VREF	VREF	
33	J2	2	Reserved	Reserved	Reserved	Reserved	
33	J2	3	Reserved	Reserved	Reserved	Reserved	
34	K2	0	GPIO202	(I/O/OD)-8 mA	VTR	VTR	
34	K2	Default: 1	ADC2	I	VREF	VREF	
34	K2	2	Reserved	Reserved	Reserved	Reserved	
34	K2	3	Reserved	Reserved	Reserved	Reserved	
35	K1	0	GPIO212	(I/O/OD)-8 mA	VTR	VTR	
35	K1	Default: 1	ADC10	I	VREF	VREF	
35	K1	2	Reserved	Reserved	Reserved	Reserved	
35	K1	3	Reserved	Reserved	Reserved	Reserved	
36	L1	0	GPIO203	(I/O/OD)-8 mA	VTR	VTR	
36	L1	Default: 1	ADC3	I	VREF	VREF	
36	L1	2	Reserved	Reserved	Reserved	Reserved	
36	L1	3	Reserved	Reserved	Reserved	Reserved	
37	K3	0	GPIO213	(I/O/OD)-8 mA	VTR	VTR	
37	K3	Default: 1	ADC11	I	VREF	VREF	
37	K3	2	Reserved	Reserved	Reserved	Reserved	
37	K3	3	Reserved	Reserved	Reserved	Reserved	
38	J4		AVTR_ADC	PWR	PWR		
38	J4						
38	J4						
38	J4						
39	K4	0	GPIO204	(I/O/OD)-8 mA	VTR	VTR	
39	K4	Default: 1	ADC4	I	VREF	VREF	
39	K4	2	Reserved	Reserved	Reserved	Reserved	
39	K4	3	Reserved	Reserved	Reserved	Reserved	
40	L2	0	GPIO214	(I/O/OD)-8 mA	VTR	VTR	
40	L2	Default: 1	ADC12	I	VREF	VREF	
40	L2	2	Reserved	Reserved	Reserved	Reserved	
40	L2	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-30: MULTIPLEXING TABLE (6 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
41	M1	0	GPIO205	(I/O/OD)-8 mA	VTR	VTR	
41	M1	Default: 1	ADC5	I	VREF	VREF	
41	M1	2	Reserved	Reserved	Reserved	Reserved	
41	M1	3	Reserved	Reserved	Reserved	Reserved	
42	M2	0	GPIO215	(I/O/OD)-8 mA	VTR	VTR	
42	M2	Default: 1	ADC13	I	VREF	VREF	
42	M2	2	Reserved	Reserved	Reserved	Reserved	
42	M2	3	Reserved	Reserved	Reserved	Reserved	
43	L3	0	GPIO206	(I/O/OD)-8 mA	VTR	VTR	
43	L3	Default: 1	ADC6	I	VREF	VREF	
43	L3	2	Reserved	Reserved	Reserved	Reserved	
43	L3	3	Reserved	Reserved	Reserved	Reserved	
44	L4	0	GPIO216	(I/O/OD)-8 mA	VTR	VTR	
44	L4	Default: 1	ADC14	I	VREF	VREF	
44	L4	2	Reserved	Reserved	Reserved	Reserved	
44	L4	3	Reserved	Reserved	Reserved	Reserved	
45	M3	0	GPIO207	(I/O/OD)-8 mA	VTR	VTR	
45	M3	Default: 1	ADC7	I	VREF	VREF	
45	M3	2	Reserved	Reserved	Reserved	Reserved	
45	M3	3	Reserved	Reserved	Reserved	Reserved	
46	M4	0	GPIO217	(I/O/OD)-8 mA	VTR	VTR	
46	M4	Default: 1	ADC15	I	VREF	VREF	
46	M4	2	Reserved	Reserved	Reserved	Reserved	
46	M4	3	Reserved	Reserved	Reserved	Reserved	
47	J5		VSS_ADC	PWR	PWR	PWR	
47	J5						
47	J5						
47	J5						
48	L5	0	Reserved	Reserved	Reserved	Reserved	
48	L5	Default: 1	LRESET#	PCI_I	VTR	VCC	Note 12
48	L5	2	Reserved	Reserved	Reserved	Reserved	
48	L5	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-31: MULTIPLEXING TABLE (7 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
49	K5	0	Reserved	Reserved	Reserved	Reserved	
49	K5	Default: 1	CLKRUN#	PCI_OD	VTR	VCC	Note 12
49	K5	2	Reserved	Reserved	Reserved	Reserved	
49	K5	3	Reserved	Reserved	Reserved	Reserved	
50	K6	0	Reserved	Reserved	Reserved	Reserved	
50	K6	Default: 1	LFRAME#	PCI_I	VTR	VCC	Note 12
50	K6	2	Reserved	Reserved	Reserved	Reserved	
50	K6	3	Reserved	Reserved	Reserved	Reserved	
51	J6	0	Reserved	Reserved	Reserved	Reserved	
51	J6	Default: 1	LDRQ#	PCI_O-8 mA	VTR	VCC	Note 12
51	J6	2	Reserved	Reserved	Reserved	Reserved	
51	J6	3	Reserved	Reserved	Reserved	Reserved	
52	M5	0	Reserved	Reserved	Reserved	Reserved	
52	M5	Default: 1	SER_IRQ	PCI_IO	VTR	VCC	Note 12
52	M5	2	Reserved	Reserved	Reserved	Reserved	
52	M5	3	Reserved	Reserved	Reserved	Reserved	
53	H7		VTR	PWR	PWR	PWR	
53	H7						
53	H7						
53	H7						
54	M6	0	Reserved	Reserved	Reserved	Reserved	
54	M6	Default: 1	PCI_CLK	PCI_ICLK	VTR	VCC	Note 12
54	M6	2	Reserved	Reserved	Reserved	Reserved	
54	M6	3	Reserved	Reserved	Reserved	Reserved	
55	L6	Default: 0	LAD0	PCI_IO	VTR	VCC	
55	L6	1	Reserved	Reserved	Reserved	Reserved	
55	L6	2	Reserved	Reserved	Reserved	Reserved	
55	L6	3	Reserved	Reserved	Reserved	Reserved	
56	L7	Default: 0	LAD1	PCI_IO	VTR	VCC	
56	L7	1	Reserved	Reserved	Reserved	Reserved	
56	L7	2	Reserved	Reserved	Reserved	Reserved	
56	L7	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-32: MULTIPLEXING TABLE (8 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
57	K7	Default: 0	LAD2	PCI_IO	VTR	VCC	
57	K7	1	Reserved	Reserved	Reserved	Reserved	
57	K7	2	Reserved	Reserved	Reserved	Reserved	
57	K7	3	Reserved	Reserved	Reserved	Reserved	
58	J7	Default: 0	LAD3	PCI_IO	VTR	VCC	
58	J7	1	Reserved	Reserved	Reserved	Reserved	
58	J7	2	Reserved	Reserved	Reserved	Reserved	
58	J7	3	Reserved	Reserved	Reserved	Reserved	
59	J8	Default: 0	GPIO100	(I/O/OD)-8 mA	VTR	VTR	
59	J8	1	nEC_SCI	OD-8 mA	VTR	VTR	
59	J8	2	Reserved	Reserved	Reserved	Reserved	
59	J8	3	Reserved	Reserved	Reserved	Reserved	
60	M7	Default: 0	GPIO011	(I/O/OD)-12 mA	VTR	VTR	
60	M7	1	nSMI	OD-12 mA	VTR	VTR	
60	M7	2	Reserved	Reserved	Reserved	Reserved	
60	M7	3	Reserved	Reserved	Reserved	Reserved	
61	K8	Default: 0	GPIO061	(I/O/OD)-8 mA	VTR	VTR	
61	K8	1	LPCPD#	I	VTR	VCC	
61	K8	2	Reserved	Reserved	Reserved	Reserved	
61	K8	3	Reserved	Reserved	Reserved	Reserved	
62	M8	Default: 0	nFWP	I	VTR	VTR	
62	M8	1	Reserved	Reserved	Reserved	Reserved	
62	M8	2	Reserved	Reserved	Reserved	Reserved	
62	M8	3	Reserved	Reserved	Reserved	Reserved	
63	L8	Default: 0	GPIO050	(I/O/OD)-8 mA	VTR	VTR	
63	L8	1	FAN_TACH0	I	VTR	VTR	
63	L8	2	Reserved	Reserved	Reserved	Reserved	
63	L8	3	Reserved	Reserved	Reserved	Reserved	
64	M9	Default: 0	GPIO051	(I/O/OD)-8 mA	VTR	VTR	
64	M9	1	FAN_TACH1	I	VTR	VTR	
64	M9	2	Reserved	Reserved	Reserved	Reserved	
64	M9	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-33: MULTIPLEXING TABLE (9 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
65	L9	Default: 0	GPIO052	(I/O/OD)-8 mA	VTR	VTR	
65	L9	1	FAN_TACH2	I	VTR	VTR	
65	L9	2	Reserved	Reserved	Reserved	Reserved	
65	L9	3	Reserved	Reserved	Reserved	Reserved	
66	K9	Default: 0	GPIO016	(I/O/OD)-8 mA	VTR	VTR	
66	K9	1	GPTP-IN7	I	VTR	VTR	
66	K9	2	FAN_TACH3	I	VTR	VTR	
66	K9	3	Reserved	Reserved	Reserved	Reserved	
67	L10	Default: 0	GPIO053	(I/O/OD)-12 mA	VTR	VTR	
67	L10	1	PWM0	(O/OD)-12 mA	VTR	VTR	
67	L10	2	Reserved	Reserved	Reserved	Reserved	
67	L10	3	Reserved	Reserved	Reserved	Reserved	
68	M10	Default: 0	GPIO054	(I/O/OD)-12 mA	VTR	VTR	
68	M10	1	PWM1	(O/OD)-12 mA	VTR	VTR	
68	M10	2	Reserved	Reserved	Reserved	Reserved	
68	M10	3	Reserved	Reserved	Reserved	Reserved	
69	M11	Default: 0	GPIO055	(I/O/OD)-12 mA	VTR	VTR	
69	M11	1	PWM2	(O/OD)-12 mA	VTR	VTR	
69	M11	2	Reserved	Reserved	Reserved	Reserved	
69	M11	3	Reserved	Reserved	Reserved	Reserved	
70	L11	Default: 0	GPIO056	(I/O/OD)-12 mA	VTR	VTR	
70	L11	1	PWM3	(O/OD)-12 mA	VTR	VTR	
70	L11	2	Reserved	Reserved	Reserved	Reserved	
70	L11	3	Reserved	Reserved	Reserved	Reserved	
71	M12	Default: 0	GPIO001	(I/O/OD)-16 mA	VTR	VTR	
71	M12	1	PWM4	(O/OD)-16 mA	VTR	VTR	
71	M12	2	Reserved	Reserved	Reserved	Reserved	
71	M12	3	Reserved	Reserved	Reserved	Reserved	
72	K10	Default: 0	GPIO002	(I/O/OD)-16 mA	VTR	VTR	
72	K10	1	PWM5	(O/OD)-16 mA	VTR	VTR	
72	K10	2	Reserved	Reserved	Reserved	Reserved	
72	K10	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-34: MULTIPLEXING TABLE (10 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
73	L12	Default: 0	GPIO014	(I/O/OD)-16 mA	VTR	VTR	
73	L12	1	GPTP-IN6	I	VTR	VTR	
73	L12	2	PWM6	(O/OD)-16 mA	VTR	VTR	
73	L12	3	Reserved	Reserved	Reserved	Reserved	
74	K12	Default: 0	GPIO015	(I/O/OD)-16 mA	VTR	VTR	
74	K12	1	GPTP-OUT6	(O/OD)-16 mA	VTR	VTR	
74	K12	2	PWM7	(O/OD)-16 mA	VTR	VTR	
74	K12	3	Reserved	Reserved	Reserved	Reserved	
75	F8	Default: 0	GPIO151	(I/O/OD)-8 mA	VTR	VTR	
75	F8	1	GPTP-IN3	I	VTR	VTR	
75	F8	2	ICT4	I	VTR	VTR	
75	F8	3	KSO15	OD-8 mA	VTR	VTR	
76	E9	Default: 0	GPIO152	(I/O/OD)-8 mA	VTR	VTR	
76	E9	1	GPTP-OUT3	(O/OD)-8 mA	VTR	VTR	
76	E9	2	ICT5	I	VTR	VTR	
76	E9	3	KSO16	OD-8 mA	VTR	VTR	
77	J9		VTR	PWR	PWR	PWR	
77	J9						
77	J9						
77	J9						
78	J10	Default: 0	GPIO003	(I/O/OD)-12 mA	VTR	VTR	
78	J10	1	SMB00_DATA	(I/OD)-12 mA	VTR	VTR	
78	J10	2	Reserved	Reserved	Reserved	Reserved	
78	J10	3	Reserved	Reserved	Reserved	Reserved	
79	H10	Default: 0	GPIO004	(I/O/OD)-12 mA	VTR	VTR	
79	H10	1	SMB00_CLK	(I/OD)-12 mA	VTR	VTR	
79	H10	2	Reserved	Reserved	Reserved	Reserved	
79	H10	3	Reserved	Reserved	Reserved	Reserved	
80	K11	Default: 0	GPIO005	(I/O/OD)-12 mA	VTR	VTR	
80	K11	1	SMB01_DATA	(I/OD)-12 mA	VTR	VTR	
80	K11	2	Reserved	Reserved	Reserved	Reserved	
80	K11	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-35: MULTIPLEXING TABLE (11 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
81	J12	Default: 0	GPIO006	(I/O/OD)-12 mA	VTR	VTR	
81	J12	1	SMB01_CLK	(I/OD)-12 mA	VTR	VTR	
81	J12	2	Reserved	Reserved	Reserved	Reserved	
81	J12	3	Reserved	Reserved	Reserved	Reserved	
82	J11	Default: 0	GPIO012	(I/O/OD)-12 mA	VTR	VTR	
82	J11	1	SMB07_DATA	(I/OD)-12 mA	VTR	VTR	
82	J11	2	SMB22_DATA	(I/OD)-12 mA	VTR	VTR	
82	J11	3	Reserved	Reserved	Reserved	Reserved	
83	H11	Default: 0	GPIO013	(I/O/OD)-12 mA	VTR	VTR	
83	H11	1	SMB07_CLK	(I/OD)-12 mA	VTR	VTR	
83	H11	2	SMB22_CLK	(I/OD)-12 mA	VTR	VTR	
83	H11	3	Reserved	Reserved	Reserved	Reserved	
84	H12	Default: 0	GPIO130	(I/O/OD)-12 mA	VTR	VTR	
84	H12	1	SMB12_DATA	(I/OD)-12 mA	VTR	VTR	
84	H12	2	Reserved	Reserved	Reserved	Reserved	
84	H12	3	Reserved	Reserved	Reserved	Reserved	
85	G10	Default: 0	GPIO131	(I/O/OD)-12 mA	VTR	VTR	
85	G10	1	SMB12_CLK	(I/OD)-12 mA	VTR	VTR	
85	G10	2	Reserved	Reserved	Reserved	Reserved	
85	G10	3	Reserved	Reserved	Reserved	Reserved	
86	G11	Default: 0	GPIO132	(I/O/OD)-12 mA	VTR	VTR	
86	G11	1	SMB06_DATA	(I/OD)-12 mA	VTR	VTR	
86	G11	2	KSO14	OD-12 mA	VTR	VTR	
86	G11	3	Reserved	Reserved	Reserved	Reserved	
87	G12	Default: 0	GPIO140	(I/O/OD)-12 mA	VTR	VTR	
87	G12	1	SMB06_CLK	(I/OD)-12 mA	VTR	VTR	
87	G12	2	Reserved	Reserved	Reserved	Reserved	
87	G12	3	Reserved	Reserved	Reserved	Reserved	
88	H9		VTR_FLASH	PWR	PWR	PWR	
88	H9						
88	H9						
88	H9						

TABLE 2-36: MULTIPLEXING TABLE (12 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
89	G9	Default: 0	GPIO141	(I/O/OD)-16 mA	VTR	VTR	
89	G9	1	SMB05_DATA	(I/OD)-16 mA	VTR	VTR	
89	G9	2	SMB20_DATA	(I/OD)-16 mA	VTR	VTR	
89	G9	3	FLSCLK	O-16 mA	VTR	VTR	
90	H8	Default: 0	GPIO142	(I/O/OD)-16 mA	VTR	VTR	
90	H8	1	SMB05_CLK	(I/OD)-16 mA	VTR	VTR	
90	H8	2	SMB20_CLK	(I/OD)-16 mA	VTR	VTR	
90	H8	3	FLSOUT	(I/O/OD)-16 mA	VTR	VTR	
91	G8	Default: 0	GPIO143	(I/O/OD)-16 mA	VTR	VTR	
91	G8	1	SMB04_DATA	(I/OD)-16 mA	VTR	VTR	
91	G8	2	Reserved	Reserved	Reserved	Reserved	
91	G8	3	FLSIN	(I/O/OD)-16 mA	VTR	VTR	
92	F9	Default: 0	GPIO144	(I/O/OD)-16 mA	VTR	VTR	
92	F9	1	SMB04_CLK	(I/OD)-16 mA	VTR	VTR	
92	F9	2	Reserved	Reserved	Reserved	Reserved	
92	F9	3	FLSCS	O-16 mA	VTR	VTR	
93	F11	Default: 0	GPIO007	(I/O/OD)-12 mA	VTR	VTR	
93	F11	1	SMB03_DATA	(I/OD)-12 mA	VTR	VTR	
93	F11	2	PS2_CLK0B	(I/OD)-12 mA	VTR	VTR	
93	F11	3	Reserved	Reserved	Reserved	Reserved	
94	F10	Default: 0	GPIO010	(I/O/OD)-12 mA	VTR	VTR	
94	F10	1	SMB03_CLK	(I/OD)-12 mA	VTR	VTR	
94	F10	2	PS2_DAT0B	(I/OD)-12 mA	VTR	VTR	
94	F10	3	Reserved	Reserved	Reserved	Reserved	
95	F12	Default: 0	GPIO154	(I/O/OD)-12 mA	VTR	VTR	
95	F12	1	SMB02_DATA	(I/OD)-12 mA	VTR	VTR	
95	F12	2	PS2_CLK1B	(I/OD)-12 mA	VTR	VTR	
95	F12	3	Reserved	Reserved	Reserved	Reserved	
96	E12	Default: 0	GPIO155	(I/O/OD)-12 mA	VTR	VTR	
96	E12	1	SMB02_CLK	(I/OD)-12 mA	VTR	VTR	
96	E12	2	PS2_DAT1B	(I/OD)-12 mA	VTR	VTR	
96	E12	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-37: MULTIPLEXING TABLE (13 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
97	E11	Default: 0	GPIO110	(I/O/OD)-16 mA	VTR	VTR	
97	E11	1	PS2_CLK2	(I/OD)-16 mA	VTR	VTR	
97	E11	2	GPTP-IN5	I	VTR	VTR	
97	E11	3	Reserved	Reserved	Reserved	Reserved	
98	E10	Default: 0	GPIO111	(I/O/OD)-16 mA	VTR	VTR	
98	E10	1	PS2_DAT2	(I/OD)-16 mA	VTR	VTR	
98	E10	2	GPTP-OUT5	(O/OD)-16 mA	VTR	VTR	
98	E10	3	Reserved	Reserved	Reserved	Reserved	
99	D12	Default: 0	GPIO112	(I/O/OD)-12 mA	VTR	VTR	
99	D12	1	PS2_CLK1A	(I/O/OD)-12 mA	VTR	VTR	
99	D12	2	KSO5	OD-12 mA	VTR	VTR	
99	D12	3	Reserved	Reserved	Reserved	Reserved	
100	D11	Default: 0	GPIO113	(I/O/OD)-12 mA	VTR	VTR	
100	D11	1	PS2_DAT1A	(I/O/OD)-12 mA	VTR	VTR	
100	D11	2	KSO6	OD-12 mA	VTR	VTR	
100	D11	3	Reserved	Reserved	Reserved	Reserved	
101	C12	Default: 0	GPIO114	(I/O/OD)-12 mA	VTR	VTR	
101	C12	1	PS2_CLK0A	(I/OD)-12 mA	VTR	VTR	
101	C12	2	Reserved	Reserved	Reserved	Reserved	
101	C12	3	Reserved	Reserved	Reserved	Reserved	
102	D10	Default: 0	GPIO115	(I/O/OD)-12 mA	VTR	VTR	
102	D10	1	PS2_DAT0A	(I/OD)-12 mA	VTR	VTR	
102	D10	2	Reserved	Reserved	Reserved	Reserved	
102	D10	3	Reserved	Reserved	Reserved	Reserved	
103	C11	Default: 0	GPIO145	(I/O/OD)-12 mA	VTR	VTR	
103	C11	1	SMB11_DATA	(I/OD)-12 mA	VTR	VTR	
103	C11	2	JTAG_TDI	I	VTR	VTR	
103	C11	3	Reserved	Reserved	Reserved	Reserved	
104	B12	Default: 0	GPIO146	(I/O/OD)-12 mA	VTR	VTR	
104	B12	1	SMB11_CLK	(I/OD)-12 mA	VTR	VTR	
104	B12	2	JTAG_TDO	O-12 mA	VTR	VTR	
104	B12	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-38: MULTIPLEXING TABLE (14 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
105	B11	Default: 0	GPIO147	(I/O/OD)-12 mA	VTR	VTR	
105	B11	1	SMB10_DATA	(I/OD)-12 mA	VTR	VTR	
105	B11	2	SMB21_DATA	(I/OD)-12 mA	VTR	VTR	
105	B11	3	JTAG_CLK	I	VTR	VTR	
106	A12	Default: 0	GPIO150	(I/O/OD)-12 mA	VTR	VTR	
106	A12	1	SMB10_CLK	(I/OD)-12 mA	VTR	VTR	
106	A12	2	SMB21_CLK	(I/OD)-12 mA	VTR	VTR	
106	A12	3	JTAG_TMS	I	VTR	VTR	
107	A11	Default: 0	JTAG_RST#	I	VTR	VTR	
107	A11	1	Reserved	Reserved	Reserved	Reserved	
107	A11	2	Reserved	Reserved	Reserved	Reserved	
107	A11	3	Reserved	Reserved	Reserved	Reserved	
108	C10	Default: 0	GPIO104	(I/O/OD)-12 mA	VTR	VTR	
108	C10	1	UART_TX	O-12 mA	VTR	(VTR/VCC)	
108	C10	2	Reserved	Reserved	Reserved	Reserved	
108	C10	3	Reserved	Reserved	Reserved	Reserved	
109	A10	Default: 0	GPIO105	(I/O/OD)-8 mA	VTR	VTR	
109	A10	1	UART_RX	I	VTR	VTR	
109	A10	2	Reserved	Reserved	Reserved	Reserved	
109	A10	3	Reserved	Reserved	Reserved	Reserved	
110	B10	Default: 0	GPIO025	(I/O/OD)-8 mA	VTR	VTR	
110	B10	1	UART_CLK	I	VTR	VTR	
110	B10	2	TIN0	I	VTR	VTR	
110	B10	3	EM_INT	(O/OD)-8 mA	VTR	VCC	
111	B9	Default: 0	GPIO026	(IS/O/OD)-8 mA	VTR	VTR	
111	B9	1	GPTP-IN0	IS	VTR	VTR	
111	B9	2	TIN1	IS	VTR	VTR	
111	B9	3	KSI3	IS	VTR	VTR	
112	D9	Default: 0	GPIO027	(IS/O/OD)-8 mA	VTR	VTR	
112	D9	1	GPTP-OUT0	(O/OD)-8 mA	VTR	VTR	
112	D9	2	TIN2	IS	VTR	VTR	
112	D9	3	KSI4	IS	VTR	VTR	

TABLE 2-39: MULTIPLEXING TABLE (15 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
113	C9	Default: 0	GPIO030	(IS/O/OD)-8 mA	VTR	VTR	
113	C9	1	GPTP-IN1	IS	VTR	VTR	
113	C9	2	TIN3	IS	VTR	VTR	
113	C9	3	KSI5	IS	VTR	VTR	
114	A9	Default: 0	GPIO107	(I/O/OD)-8 mA	VTR	VTR	
114	A9	1	KSO4	OD-8 mA	VTR	VTR	
114	A9	2	Reserved	Reserved	Reserved	Reserved	
114	A9	3	Reserved	Reserved	Reserved	Reserved	
115	D8	Default: 0	GPIO120	(I/O/OD)-8 mA	VTR	VTR	
115	D8	1	KSO7	OD-8 mA	VTR	VTR	
115	D8	2	Reserved	Reserved	Reserved	Reserved	
115	D8	3	Reserved	Reserved	Reserved	Reserved	
116	B8	Default: 0	GPIO124	(I/O/OD)-8 mA	VTR	VTR	
116	B8	1	GPTP-OUT4	(O/OD)-8 mA	VTR	VTR	
116	B8	2	KSO11	OD-8 mA	VTR	VTR	
116	B8	3	Reserved	Reserved	Reserved	Reserved	
117	A8	Default: 0	GPIO125	(I/O/OD)-8 mA	VTR	VTR	
117	A8	1	GPTP-IN4	I	VTR	VTR	
117	A8	2	KSO12	OD-8 mA	VTR	VTR	
117	A8	3	Reserved	Reserved	Reserved	Reserved	
118	C8	Default: 0	GPIO031	(IS/O/OD)-8 mA	VTR	VTR	
118	C8	1	GPTP-OUT1	(O/OD)-8 mA	VTR	VTR	
118	C8	2	TOUT0	O-8 mA	VTR	VTR	
118	C8	3	KSI6	IS	VTR	VTR	
119	B7	Default: 0	GPIO032	(IS/O/OD)-8 mA	VTR	VTR	
119	B7	1	GPTP-IN2	IS	VTR	VTR	
119	B7	2	TOUT1	O-8 mA	VTR	VTR	
119	B7	3	KSI7	IS	VTR	VTR	
120	D7	Default: 0	GPIO040	(I/O/OD)-8 mA	VTR	VTR	
120	D7	1	GPTP-OUT2	(O/OD)-8 mA	VTR	VTR	
120	D7	2	TOUT2	O-8 mA	VTR	VTR	
120	D7	3	KSO0	OD-8 mA	VTR	VTR	

TABLE 2-40: MULTIPLEXING TABLE (16OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
121	A7	Default: 0	GPIO017	(IS/O/OD)-8 mA	VTR	VTR	
121	A7	1	GPTP-OUT7	(O/OD)-8 mA	VTR	VTR	
121	A7	2	TOUT3	O-8 mA	VTR	VTR	
121	A7	3	KSI0	IS	VTR	VTR	
122	F6	Default: 0	GPIO022	(I/O/OD)-16 mA	VTR	VTR	
122	F6	1	BCM_B_CLK	O-16 mA	VTR	VTR	
122	F6	2	V_CLK	O-16 mA	VTR	VTR	
122	F6	3	Reserved	Reserved	Reserved	Reserved	
123	E6	Default: 0	GPIO023	(I/O/OD)-16 mA	VTR	VTR	
123	E6	1	BCM_B_DAT	IO-16 mA	VTR	VTR	
123	E6	2	V_DATA	IO-16 mA	VTR	VTR	
123	E6	3	Reserved	Reserved	Reserved	Reserved	
124	F5	Default: 0	GPIO024	(I/O/OD)-16 mA	VTR	VTR	
124	F5	1	BCM_B_INT#	IO-16 mA	VTR	VTR	
124	F5	2	V_FRAME	O-16 mA	VTR	VTR	
124	F5	3	Reserved	Reserved	Reserved	Reserved	
125	A6	Default: 0	GPIO045	(I/O/OD)-8 mA	VTR	VTR	
125	A6	1	LSBCM_D_INT#	I	VTR	VTR	
125	A6	2	KSO1	OD-8 mA	VTR	VTR	
125	A6	3	Reserved	Reserved	Reserved	Reserved	
126	C7	Default: 0	GPIO046	(I/O/OD)-8 mA	VTR	VTR	
126	C7	1	LSBCM_D_DAT	I/O-8 mA	VTR	VTR	
126	C7	2	KSO2	OD-8 mA	VTR	VTR	
126	C7	3	Reserved	Reserved	Reserved	Reserved	
127	B6	Default: 0	GPIO047	(I/O/OD)-8 mA	VTR	VTR	
127	B6	1	LSBCM_D_CLK	O-8 mA	VTR	VTR	
127	B6	2	KSO3	OD-8 mA	VTR	VTR	
127	B6	3	Reserved	Reserved	Reserved	Reserved	
128	C6	Default: 0	GPIO121	(I/O/OD)-8 mA	VTR	VTR	
128	C6	1	BCM_A_INT#	I	VTR	VTR	
128	C6	2	KSO8	OD-8 mA	VTR	VTR	
128	C6	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-41: MULTIPLEXING TABLE (17 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
129	A5	Default: 0	GPIO122	(I/O/OD)-16 mA	VTR	VTR	
129	A5	1	BCM_A_DAT	IO-16 mA	VTR	VTR	
129	A5	2	KSO9	OD-16 mA	VTR	VTR	
129	A5	3	Reserved	Reserved	Reserved	Reserved	
130	E5		VTR	PWR	PWR	PWR	
130	E5						
130	E5						
130	E5						
131	B5	Default: 0	GPIO123	(I/O/OD)-16 mA	VTR	VTR	
131	B5	1	BCM_A_CLK	O-16 mA	VTR	VTR	
131	B5	2	KSO10	OD-16 mA	VTR	VTR	
131	B5	3	Reserved	Reserved	Reserved	Reserved	
132	A4	Default: 0	GPIO041	(I/O/OD)-8 mA	VTR	VTR	
132	A4	1	PECI_REQUEST#	OD-8 mA	VTR	VTR	Note 19
132	A4	2	Reserved	Reserved	Reserved	Reserved	
132	A4	3	Reserved	Reserved	Reserved	Reserved	
133	A3	Default: 0	GPIO042	(I/O/OD)-8 mA	VTR	VTR	
133	A3	1	BCM_C_INT#	I	VTR	VTR	
133	A3	2	PECI_DAT	IO-8 mA	VREF_PECI	VREF_PECI	
133	A3	3	SB-TSI_DAT	(I/OD)-8 mA	VREF_PECI	VREF_PECI	
134	A2	Default: 0	GPIO043	(I/O/OD)-16 mA	VTR	VTR	
134	A2	1	BCM_C_DAT	I/O-16 mA	VTR	VTR	
134	A2	2	PECI_RDY	I	VREF_PECI	VREF_PECI	
134	A2	3	SB-TSI_CLK	(I/OD)-16 mA	VREF_PECI	VREF_PECI	
135	A1	Default: 0	GPIO044	(I/O/OD)-16 mA	VTR	VTR	
135	A1	1	BCM_C_CLK	O-16 mA	VTR	VTR	
135	A1	2	VREF_PECI	I	VREF_PECI	VREF_PECI	
135	A1	3	Reserved	PWR-16 mA	Reserved	Reserved	
136	B4	Default: 0	GPIO126	(I/O/OD)-8 mA	VTR	VTR	
136	B4	1	KSO13	OD-8 mA	VTR	VTR	
136	B4	2	Reserved	Reserved	Reserved	Reserved	
136	B4	3	Reserved	Reserved	Reserved	Reserved	

TABLE 2-42: MULTIPLEXING TABLE (18 OF 18)

Pin Ref. Number	Ball	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Notes
137	D6	Default: 0	GPIO020	(IS/O/OD)-8 mA	VTR	VTR	
137	D6	1	KSI1	IS	VTR	VTR	
137	D6	2	Reserved	Reserved	Reserved	Reserved	
137	D6	3	Reserved	Reserved	Reserved	Reserved	
138	B3	Default: 0	GPIO156	(I/O/OD)-16 mA	VTR	VTR	
138	B3	1	LED0	(IO/IOD)-16 mA	VTR	VTR	
138	B3	2	Reserved	OD-16 mA	Reserved	Reserved	
138	B3	3	Reserved	Reserved	Reserved	Reserved	
139	B2	Default: 0	GPIO157	(I/O/OD)-16 mA	VTR	VTR	
139	B2	1	LED1	(O/OD)-16 mA	VTR	VTR	
139	B2	2	Reserved	Reserved	Reserved	Reserved	
139	B2	3	Reserved	Reserved	Reserved	Reserved	
140	B1	Default: 0	GPIO153	(I/O/OD)-16 mA	VTR	VTR	
140	B1	1	LED2	(O/OD)-16 mA	VTR	VTR	
140	B1	2	Reserved	Reserved	Reserved	Reserved	
140	B1	3	Reserved	Reserved	Reserved	Reserved	
141	D5		VSS	PWR	PWR	PWR	
141	D5						
141	D5						
141	D5						
142	C3		NO_CONNECT				
142	C3						
142	C3						
142	C3						
143	C5		NO_CONNECT				
143	C5						
143	C5						
143	C5						
144	E4		NO_CONNECT				
144	E4						
144	E4						
144	E4						

2.6 Notes for Tables in this Chapter

Note 1	Buffer modes are described per signal function. On multiplexed pins buffer modes are separated by a slash "/"; e.g., a pin with two multiplexed functions where the primary function is an input and the secondary function is an 8mA bi-directional driver is represented as "I/IO-8". Buffer modes in parentheses represent multiple buffer modes for a single pin function. The number following the “-“ represents the balanced output sink/source capability of the buffer in milliamps.
Note 2	This pin function exhibits "VCC" power domain emulation. The System Runtime Supply power is not connected to the MEC1609. The VCC_PWRGD signal is used to indicate when power is applied to the System Runtime Supply. All MEC1609 inputs and output signals that require "VCC" power domain functionality are powered by VTR and controlled by the VCC_PWRGD signal input. See the Pin Default State Transitions from Powered to Unpowered Table.
Note 3	The nEC_SCI pin can be controlled by hardware and EC firmware. The nEC_SCI pin can drive either the ACPI Run-time GPE Chipset input or the Wake GPE Chipset input. Depending how the nEC_SCI pin is used, other ACPI-related SCI functions may be best supplied by other general purpose outputs that can be configured as open-drain drivers.
Note 4	These pins require an external weak pull-up resistors of 10k-100k ohms.
Note 5	These pins are 3.3V only (non-5V tolerant).
Note 6	This pin is tristated when PWRGD is inactive and the pin is configured as a VCC2-powered alternate function.
Note 7	This pin function exhibits "VTR" power domain emulation. This pin is powered by VBAT power. All MEC1609 inputs and output signals functions that require "VTR" power domain emulation functionality are powered by VBAT and controlled internally by the application of VTR power. This pin is tristated when VTR power is not applied and the pin is configured as a VTR emulation powered signal function.
Note 8	Signals on this pin can be configured to generate a wake-up event to the EC on selected edges.
Note 9	A pull-up is not needed on this BC-Link DATA pin as long as the voltage remains above the logic-high threshold during the second turnaround cycle.
Note 10	This pin may require a weak pull-up.
Note 11	Most GPIO pins are (I/O/OD). See the 'Pin Multiplexing' tables and associated notes for specific exceptions.
Note 12	This pin has EC wakeup and interrupt capability controlled by the corresponding Pin Control Register. A GPIO assignment is documented in the GPIO chapter to provide interrupt and wakeup capability. The GPIO should not be used for I/O. See Detailed Pin Multiplexing Assignments section in the GPIO chapter and lookup this pin and see the associated note.

Note 13	The two pin debug port UART can be used by the Host or EC. This pin can be VCC protected or not VCC portected under program control by the POWER bit in the Configuration Select Register in Host configuration space (also accessible by the EC).
Note 14	When the JTAG_RST# pin is not asserted (logic'1'), the JTAG_TDI, JTAG_TDO, JTAG_CLK, 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_CLK, 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 route the JTAG interface to the pins. System Board Designer should terminate this pin in all functional state using jumpers and pull-up or pull down resistors, etc .
Note 15	All VBAT powered pins with GPIO's on then have only one direction selected by the default signal function. The associated GPIO input register, output register bits are not connected to the pin. Only the Interrupt Detection field in the associated pin control register function; the remainder of the bits in the pin control register has no effect.
Note 16	PS/2 ports ending with signal functions ending with "A" or "B" are muxed to a single controller. Only one set of clock and data are intended to used at a time (either "A" or "B" not both. The unused port segment should have its associated pin control register's, Mux Control Field programmed away from the PS2 controller.
Note 17	Most GPIO's are (I/O/OD). See Multiplexing tables below and associated notes for specific exceptions.
Note 18	The GPIO assignment on this pin only provides interrupt and wakeup capability. This is provided by the Interrupt Detection field in the Pin Control register. The Mux control field in the Pin Control Register should **not** be set to '00' = GPIO or undesirable results may occur.
Note 19	The PECI REQUEST# signal function must be configured as open-drain driver with an external pullup to VCC.
Note 20	This pin is also used as a JTAG TAP controller select strap option. There is a weak pullup enabled on this pin by default.

2.7 Strapping Option

GPIO116 is used for the TAP Controller Select Strap Option (see Section 39.2.1 on page 477). If any of the MEC1609/MEC1609i JTAG TAP controllers are used, GPIO116 must only be configured as an output to a VTR powered external function. GPIO116 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 as defined in Section 39.2.2 on page 477.

2.8 Package Outlines

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

FIGURE 2-2: 144-PIN LFBGA 10X10X0.8 MM PACKAGE OUTLINE (1.4 MM HEIGHT)

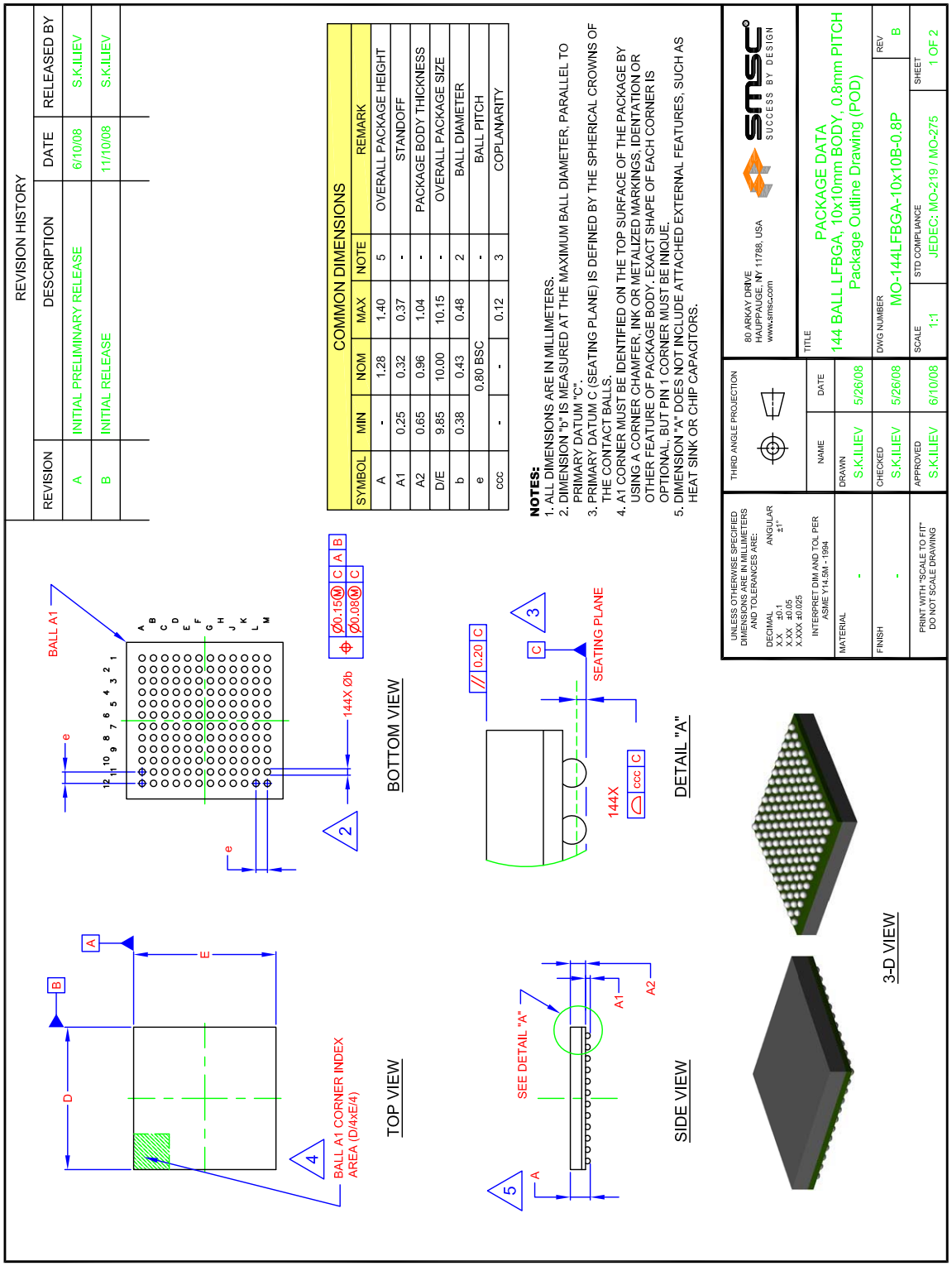
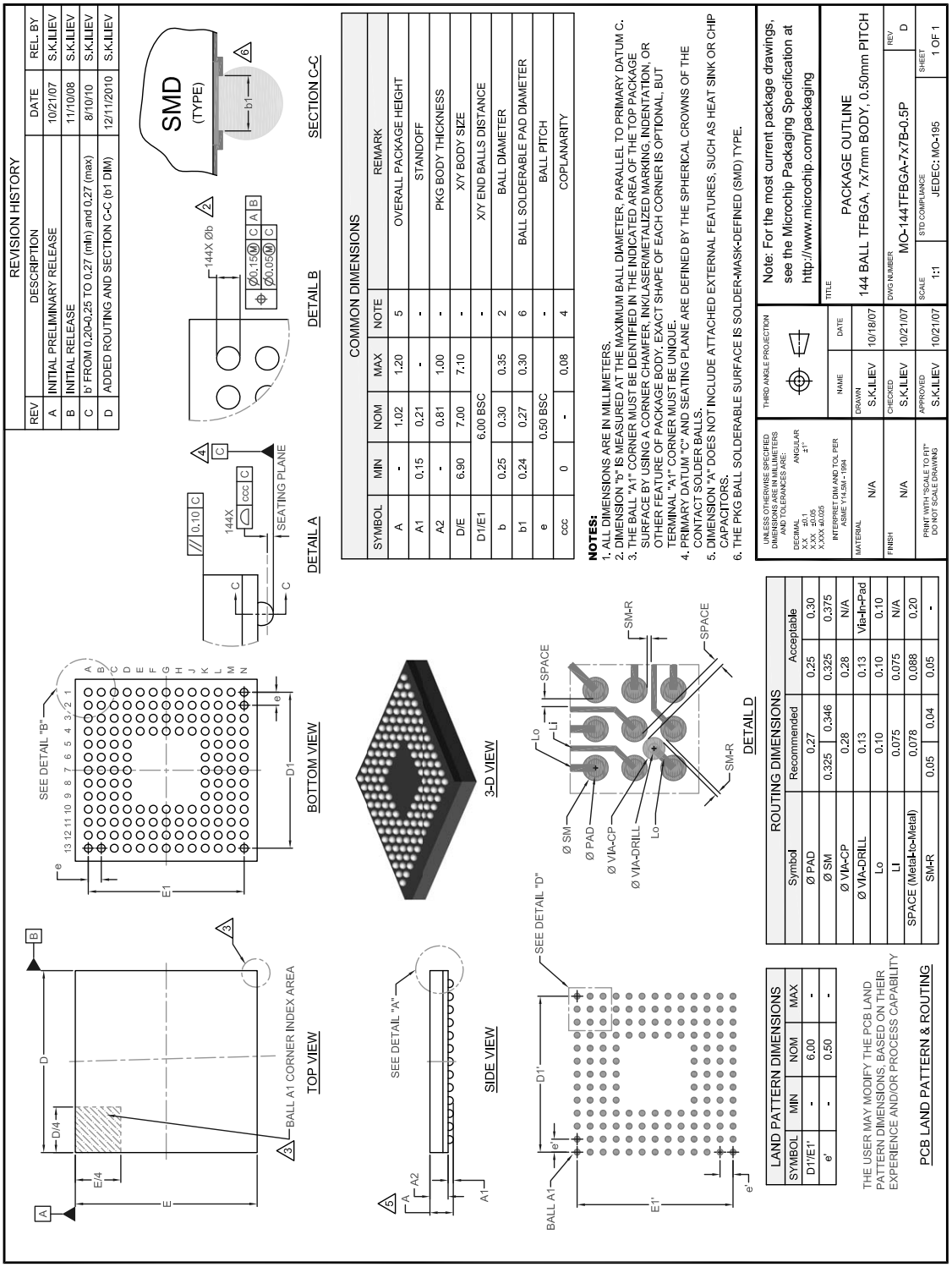


FIGURE 2-3: 144-PIN TFBGA 7X7X0.5 MM PACKAGE OUTLINE (1.2 MM HEIGHT)



3.0 BUS HIERARCHY

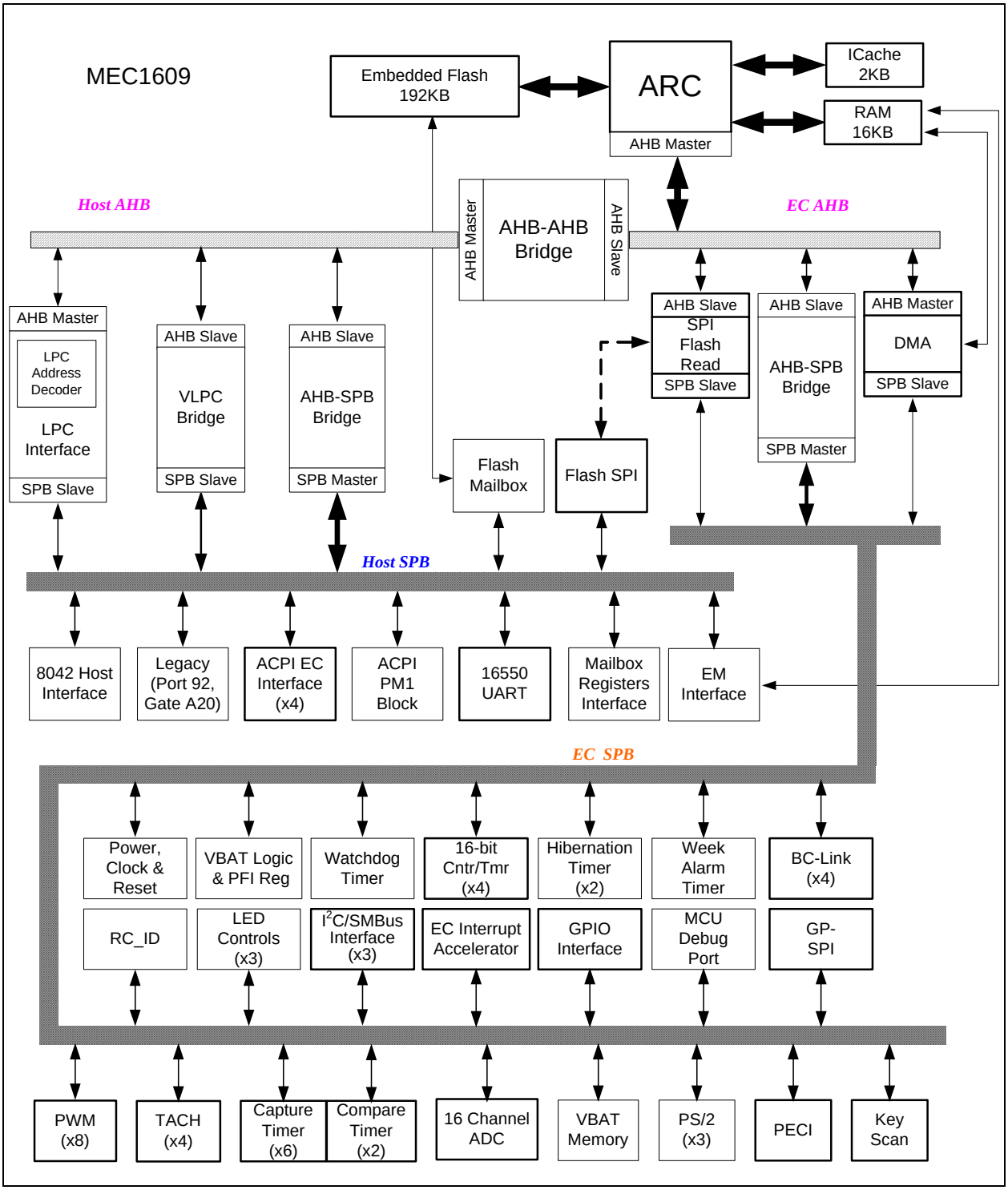
3.1 General Description

All devices in the MEC1609/MEC1609i are maintained in a common address space. All communication with on-chip functions is through registers that have addresses in this common address space. The ARC Embedded Controller (EC) can reference all devices through the address space, while the Host can only reference a subset.

3.2 Block Diagram

FIGURE 3-1: Bus Hierarchy on page 45 shows, in graphic form, the inter connectivity of devices on the MEC1609/MEC1609i, including the EC, the principal lower buses and most of the peripherals.

FIGURE 3-1: BUS HIERARCHY



3.3 Address Space

The ARC EC has a 24-bit address space. Addresses in the lower half of the range, 0h through 7F_FFFFh, can be used for both instruction access and data access. The address range 0h through 4_FFFFh (which includes the 192KB Flash Memory Array and an additional 2KB Flash Info Block) is mapped to the Embedded Flash Subsystem. Addresses in the range 80_0000h through 80_3FFFh are mapped to the Closely Coupled Data Memory. These memories are shown in Figure 3-2. Software can change the address mapping so that the Closely Coupled Data Memory is mapped to 6_0000h through 6_0FFFh, where it can be used for both instructions and data, and the Embedded Flash Subsystem becomes accessible only through a register interface. Addresses greater or equal to 80_1000h are propagated through the AHB interface on the ARC processor. References, by either the EC or the Host via the LPC bus, to addresses that are not mapped to any device register or memory will cause a bus error; see Section 3.4.3, "AHB Bus Errors," on page 52.

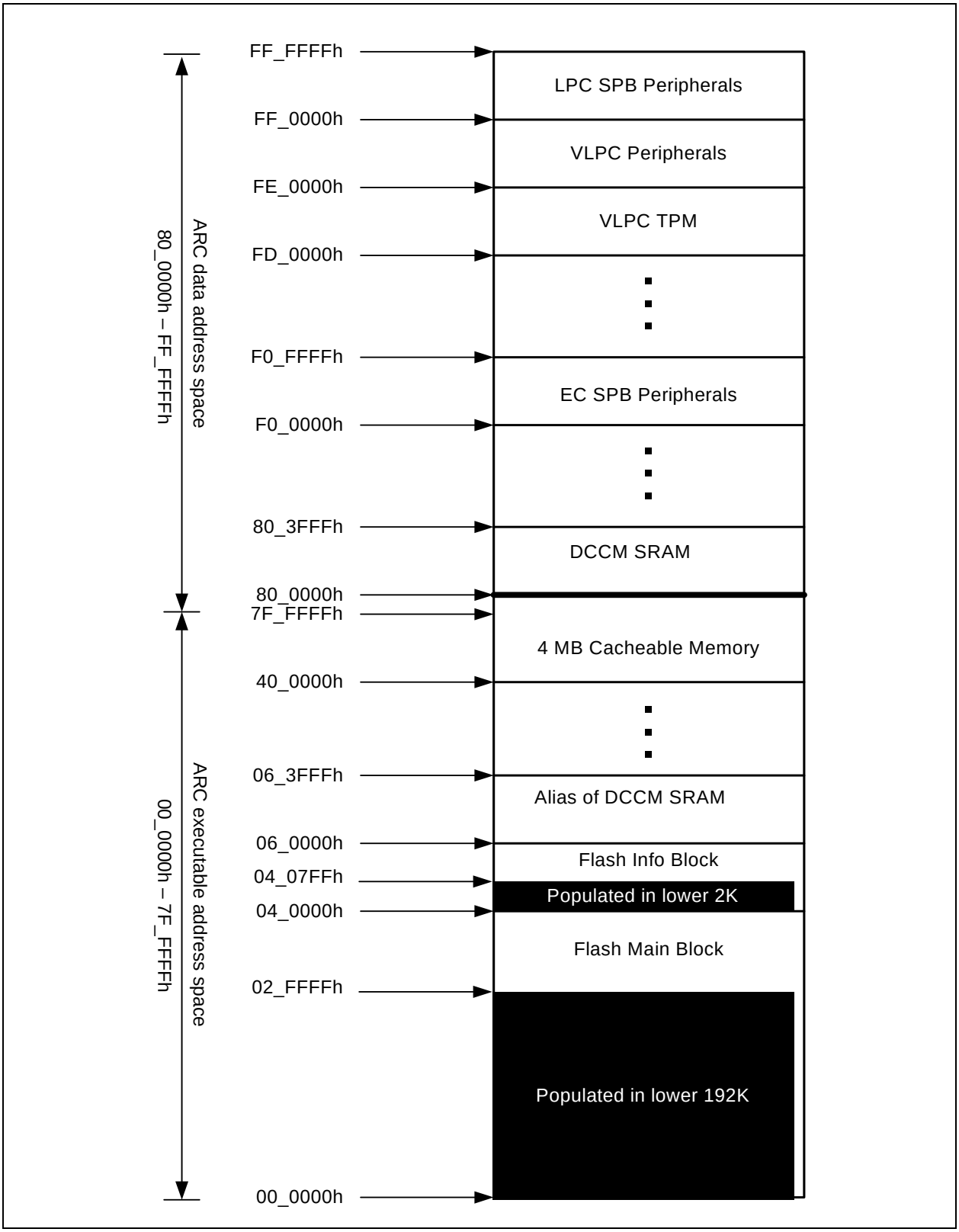
3.3.1 ARC ADDRESS SPACE

The ARC Address Space is illustrated in Figure 3-2, "MEC1609/MEC1609i EC Memory Map" & Figure 3-1, "Bus Hierarchy". EC Instruction Memory occupies the lower half of the address space, 00_0000h through 7F_FFFFh. The ARC processor can only execute instructions that are located in the Instruction Memory portion of the address space. Only a portion of the Instruction space is populated.

The upper half of the address space is used solely for data references. The region contains a general-purpose EC Data Memory SRAM. as well as the address space of the two SPB peripheral buses and the VLPC peripheral bus.

The contents of the Instruction memory can be read and written by the ARC processor through regular load and store instructions. However, there may be a one cycle instruction fetch penalty whenever a data load or store instruction to Instruction memory is executed.

FIGURE 3-2: MEC1609/MEC1609I EC MEMORY MAP



3.3.2 AHB ADDRESS SPACE

The components on the AHB subsystem and the two SPB bridges, along with the VLPC bridge, define a set of addresses that are accessible by the EC. This address space is shown in Table 3-1, "MEC1609/MEC1609i Peripheral Address Space". As shown in the table, the Host can access much of the address space, but not all.

TABLE 3-1: MEC1609/MEC1609I PERIPHERAL ADDRESS SPACE

Address Range	Device	Accessible by EC	Accessible by Host
F0_0000h - F0_FFFFh	ARC-only SPB Bridge	Yes	No
FD_0000h - FE_FFFFh	VLPC Bridge	Yes	Yes (via LPC-VLPC Link)
FF_0000h - FF_BFFFh and FF_D000h - FF_FFFFh	LPC SPB Bridge	Yes	Yes (limited by LPC interface map)
FF_C000h - FF_CFFFh	LPC AHB Bus LDN 30h	Yes	Yes (limited to 00h-FFh)

The VLPC Bridge address range of FE_0000h through FE_FFFFh is associated with the Companion devices on the VLPC bus. The address range of FD_0000h through FD_FFFFh is associated with a TPM on the VLPC bus. LPC Trusted Cycles in the address range 0000h through 4FFFh are forwarded onto the AHB bus at addresses FD_0000h through FD_4FFFh, and thus through the VLPC bridge. See Section 6.3.1, "LPC Bus Interface," on page 117.

The 64KB address space of an SPB Bridge, as well as 48KB of the address space of the VLPC Bridge, is divided into 1KB Frames. Peripherals are grouped into Logical Devices; each Logical Device corresponds to a 1KB Frame. Logical Devices addressable by the Host are listed in Table 3-2, "Host Logical Devices on MEC1609/MEC1609i". Host-accessible Logical Devices may be located in Companion devices on the VLPC Bus; the base addresses of VLPC Logical Device frames are shown in Table 3-3, "Host Logical Devices on VLPC Bus". The table includes two Logical Devices, 77h and 7Fh, that do not correspond to devices but can be used to access registers in the VLPC Global address space. The 1KB Address Frame of a Host Logical Device is divided into four subregions, as described in Section 3.4.2, "Address Framing," on page 51.

TABLE 3-2: HOST LOGICAL DEVICES ON MEC1609/MEC1609I

Logical Device Number	Logical Devices	AHB Address of Frame	Runtime Registers	Configuration Registers	EC-only Registers
0h	Mailbox Interface	FF_0000h	yes	yes	yes
1h	Keyboard Controller (8042)	FF_0400h	yes	yes	no
2h	ACPI EC Channel 0	FF_0800h	yes	yes	no
3h	ACPI EC Channel 1	FF_0C00h	yes	yes	no
4h	ACPI EC Channel 2	FF_1000h	yes	yes	no
5h	ACPI EC Channel 3	FF_1400h	yes	yes	no
6h	ACPI PM1	FF_1800h	yes	yes	no
7h	UART	FF_1C00h	yes	yes	no
8h	Legacy (Fast KB)	FF_2000h	yes	yes	no
Ch	LPC Interface	FF_3000h	no	yes	yes
Dh	VLPC Interface	FF_3400h	no	yes	no
Eh	Embedded Flash Interface	FF_3800h	yes	yes	yes
Fh	Flash SPI	FF_3C00h	yes	yes	no
10h	EM Interface	FF_4000h	yes	yes	yes
3Fh	Global Configuration	FF_FC00h	no	no	yes

TABLE 3-3: HOST LOGICAL DEVICES ON VLPC BUS

Logical Device Number	AHB Address for Base of Frame	Logical Devices
40h	FE_0000h	Companion 0, Logical Device 0
41h	FE_0400h	Companion 0, Logical Device 1
42h	FE_0800h	Companion 0, Logical Device 2
...	...	...
4Fh	FE_3C00h	Companion 0, Logical Device 15
50h	FE_4000h	Companion 1, Logical Device 0
51h	FE_4400h	Companion 0, Logical Device 1
...	...	...
5Fh	FE_7C00h	Companion 1, Logical Device 15
60h	FE_8000h	Companion 2, Logical Device 0
61h	FE_8400h	Companion 2, Logical Device 1
...	...	...
6Fh	FE_BC00h	Companion 2, Logical Device 15
77h	FE_DC00h	Global Read/Write registers in VLPC space
7Fh	FE_FC00h	Global Read-Only registers in VLPC space

The 64KB address space of the ARC-only SPB Bridge is divided into 1KB Frames. Peripherals are grouped into Logical Devices; each Logical Device corresponds to a 1KB Frame. These EC Logical Devices addressable by the ARC-only are listed in Table 3-4, "EC Logical Devices on MEC1609/MEC1609i". Multiple instantiations of the same block are in a single 1KB Frame. Each instantiation is separated by 128 bytes.

Note 3-1 All VBAT powered registers are in a Single 1KB Frame separated by 128 bytes (see Table 3-4, Logical Device Number 33h.)

Note 3-2 The VBAT-Powered Control Interface is in the Global Configuration Logical Device, Host Logical Device Number 3Fh, in EC-only space (see Section 3.4.2, "Address Framing," on page 51).

Note that the ARC-only SPB Bridge address space uses the terminology of Logical Device; however, there is no host access to these blocks.

TABLE 3-4: EC LOGICAL DEVICES ON MEC1609/MEC1609I

Logical Device Number	AHB Address for Base of Frame	Logical Devices	Notes
0h	F0_0000h	Hibernation Timer	
1h	F0_0400h	Watchdog Timer	
2h	F0_0800h	Input Capture and Compare Timer	
3h	F0_0C00h	16-bit Timer	
4h	F0_1000h	RC ID	
5h	F0_1400h	BC Bus Master	
6h	F0_1800h	SMBus	
7h	F0_1C00h	EC GP-SPI	
8h	F0_2000h	Keyscan Interface	
9h	F0_2400h	DMA	
Ah	F0_2800h	SPI Flash Read	
Bh - 15h	F0_2C00h - F0_4FFFh	Reserved	
16h	F0_5800h	PWM	

TABLE 3-4: EC LOGICAL DEVICES ON MEC1609/MEC1609I (CONTINUED)

Logical Device Number	AHB Address for Base of Frame	Logical Devices	Notes
17h	F0_5C00h	Reserved	
18h	F0_6000h	TACH	
19h	F0_6400h	PECI	
1Ah	F0_6800h	ADC	
1Bh - 20h	F0_6C00h - F0_83FFh	Reserved	
21h	F0_8400h	LED	
22h	F0_8800h	PS/2	
23h	F0_8C00h	MCU Debug Port	
24h - 2Fh	F0_9000h - F0_BCFFh	Reserved	
30h	F0_C000h	EC Interrupt Aggregator	
31h	F0_C400h	GPIOs	
32h	F0_C800h	Power, Clock & Reset (VTR PWR'ed)	
33h	F0_CC00h	Power, Clock & Reset (VBAT PWR'ed)	
	F0_CC80h	Week Alarm Timer	
	F0_CD00h	VBAT Backed Memory	
34h - 3Eh	F0_D400h - F0_FBFFh	Reserved	
3Fh	F0_FC00h	EC Test and Debug	

3.4 AHB Buses

Addresses and internal buses in the MEC1609/MEC1609i are compatible with ARM Limited's *Advanced Microprocessor Bus Architecture* (AMBA), as specified in *AMBA™ Specification (Rev 2.0),* 1999.

As seen in FIGURE 3-1: Bus Hierarchy on page 45, there are two separate AHB buses, the EC AHB and the Host AHB. The first has one master, the EC, and two slaves, while the second has two masters, the LPC interface and the AHB-AHB bridge, and four slaves. The bus connections are summarized in Table 3-5, "MEC1609/MEC1609i AHB Buses" and can be seen in Figure 3-1.

TABLE 3-5: MEC1609/MEC1609I AHB BUSES

AHB Bus	Master Interfaces	Slave Interfaces
EC AHB	EC	EC SPB bridge AHB-AHB bridge
Host AHB	LPC Interface AHB-AHB bridge	LPC SPB bridge SPI bridge VLPC bridge

The AHB-to-AHB bridge is a one-way device: addresses generated on the EC AHB can be propagated to the Host AHB bus, but addresses on the Host AHB bus cannot be propagated to the EC AHB bus. This is the reason that the Host is restricted from accessing the address range F0_0000h to F0_FFFFh (the address range of the EC SPB bridge). The bridge maps address from FD_0000h through FF_FFFFh. Multi-word block transfers on the EC AHB (for example, the 16-byte memory fetch to fill a cache line) are converted by the bridge into a series of sequential 4-byte transfers.

Both the EC and the LPC interface can have at most one outstanding AHB bus request at one time. On the Host AHB, the LPC interface always has priority over the AHB-AHB bridge. Both AHB buses support byte, halfword and word AHB transfers. The only Burst mode supported on the EC AHB is an incrementing burst of 4 beats of 4 bytes per beat. Burst mode is not supported on the Host AHB. AHB bus locking or early burst termination are not supported.

3.4.1 BUS CLOCKING

The Host AHB runs at the MEC1609/MEC1609i system clock rate of 64.52 MHz. The bus clock and the bus arbiter will be shut down when there are no transactions active on the bus. The bus clock and arbiter will be restarted as soon as an address is acquired from the LPC bus, or when an EC AHB bus transaction is mapped, via the AHB-to-AHB bridge, to the address space of the Host AHB. The EC AHB clock can be programmed to run at any of the available rates between 4.3 MHz and 21.5MHz (32.2 MHz if the code is aligned). See Section 5.0, "Power, Clocks and Resets," on page 73. The EC bus clock is shut down when the EC is idle.

If the Host AHB clock is idle when an LPC transaction arrives at the LPC interface, the LPC interface will restart the AHB bus clock and arbiter early enough so that as soon as an I/O address is translated to an AHB address the I/O transaction can be placed on the AHB without delay. If the I/O address is not claimed by the MEC1609/MEC1609i then the LPC interface will drop its bus request. If the EC is not requesting the Host AHB at the same time, the Host AHB bus clock will again shut down.

3.4.2 ADDRESS FRAMING

The EC can directly address all peripherals on the MEC1609/MEC1609i. The Host, by contrast, is restricted in what it can address. The Host accesses the MEC1609/MEC1609i through the LPC bus, using I/O cycles, Memory Cycles and Firmware Hub cycles (see Section 6.0, "Host Interface," on page 113). These cycles are mapped into the MEC1609/MEC1609i address space accessible by the LPC interface.

The mapping function forces some of the address bits to preset values, as shown in Table 3-6, "LPC to MEC1609/MEC1609i Address Mapping". This mapping has the effect of creating four contiguous 256-byte regions: LPC I/O, EC-only, LPC DMA and LPC Configuration. Together, the four regions create a 1024 byte “frame” for each logical device that is accessible to the Host. The mapping is further constrained to either the 64KB space of the LPC SPB or the 64KB space of the VLPC Bus. There is therefore a maximum of 64 Logical Devices, each with a 1KB frame, located on the basechip. Because there is a maximum of three Companion devices possible on the VLPC Bus, there is a maximum of 48 Logical Devices, 16 per Companion, located on Companion devices attached to the VLPC Bus.

LPC I/O cycles for Runtime Registers are mapped into the first 256 bytes of the 1024 byte frame. Configuration Registers, which are accessed through a Configuration portal typically located at addresses 2Eh and 2Fh in the LPC I/O space, are restricted to the highest 256 bytes of the 1024 byte frame.

DMA FIFO addresses are restricted to offsets between 200h and 2FFh within a Logical Device frame. In addition, DMA FIFO addresses are restricted to 32-bit aligned addresses, that is, address bits [1:0] are both 0. The low 10 bits of a DMA FIFO are thus in the form ‘10xxxxxx00b’.

LPC Firmware Hub cycles and LPC Memory cycles are mapped into the address range 80_0000h to FF_FFFFh. This range is mapped into the off-chip SPI Flash memory.

Because the EC does not require the mapping mechanism required for translating LPC Runtime Registers, Configuration Registers and DMA channels, it accesses each 1KB frame uniformly. All Logical Devices located on the EC-only AHB (those in the address range F0_0000h through F0_FFFFh) have a flat, 1KB frame. Because there is no separate EC-only address space on the VLPC bus, all Logical Devices on VLPC Bus Companions can be accessed by the Host over the LPC bus. However, if it is necessary to restrict Host access to a Companion Logical Device, that device’s registers may be located in the 256-byte quadrant in every frame for which there is no Host mapping available (that is, at frame offsets 100h through 1FFh).

For details on LPC address mapping to the MEC1609/MEC1609i address space, see Section 6.0, "Host Interface," on page 113.

TABLE 3-6: LPC TO MEC1609/MEC1609I ADDRESS MAPPING

Type of Access	Address Bit Mapping
LPC I/O Access Runtime Registers	Address bits[9:8] = 00b Address bits[23:10] set from map
No LPC Access EC-only registers	Address bits[9:8] = 01b
LPC I/O Access through Configuration Access Port Configuration Registers	Address bits[9:8] = 11b Address bits[23:10] set from map
LPC DMA Access	Address bits[9:8] = 10b Address bits[23:10] set from map

3.4.3 AHB BUS ERRORS

AHB bus requests by both the Host, through the LPC bus, and the EC, can be terminated with an AHB bus error. The handling of bus errors by the EC is described in the *ARCompact™ Instruction Set Architecture: Programmer's Reference*.

Bus errors may be caused by:

- EC I/O requests that lie outside the range of the Data Closely Coupled Memory, the EC SPB or the AHB-AHB bridge
- I/O requests to either the Host SPB or the EC SPB that map to a non-existent Logical Device
- I/O requests to an invalid register address within a valid Logical Device on either the EC SPB or the Host SPB
- I/O requests to addresses on the VLPC bus that are terminated with a bus error on the VLPC bus

3.5 Peripheral Buses (SPB)

The Peripheral Bus (SPB) is a byte-addressable bus with a 16-bit address space and a 32-bit data path. All accesses must be aligned to the data-size boundaries. The SPB supports 32-bit, 16-bit and 8-bit accesses. All peripheral accesses are 32-bits wide, so that if a peripheral cannot transfer more than 8 bits of a register in one I/O access, register addresses should all be on 32-bit boundaries even though the upper bits (bits 31 through 8) are always zero. The SPB will not assemble a 32-bit word out of multiple 8-bit accesses.

SPB transfers take two cycles, so that a read or write on the AHB to a register on the SPB will take a total of three cycles. The SPB contains an 8-bit word address, four byte lane strobes and up to 64 logical device select strobes. Data on the SPB read and write buses are always 32-bit aligned, with the byte strobes indicating which byte lane or lanes should be active during a transaction. An SPB peripheral must steer bytes, based on the byte strobes, to the correct byte lane.

There are two SPB bridges forwarded to two SPB (busses):

LPC SPB

The LPC SPB address range is FF_0000h - FF_BFFFh and FF_C400h - FF_FFFFh.

EC SPB

The EC SPB address range is F0_0000h through F0_FFFFh.

4.0 LOGICAL DEVICE CONFIGURATION

4.1 Description

The Configuration of the MEC1609/MEC1609i is very flexible and is based on the configuration architecture implemented in typical Plug-and-Play components.

The MEC1609/MEC1609i is designed for motherboard designs in which the resources required by their components are known. With its flexible resource allocation architecture, the MEC1609/MEC1609i allows the BIOS to assign resources at POST.

4.2 Location of Configuration Registers

Configuration Registers for Logical Devices accessible by the Host are located on the LPC SPB in the address range FF_0000h through FF_FFFFh. All Configuration Registers are located at addresses where address bits 9 and 8 are both '1b' (that is, at offsets 300h through 3FFh from the base of a 1KB address frame). Configuration registers are accessible by the Embedded Controller with 8-bit, 16-bit or 32-bit accesses. The Host can access the registers only with 8-bit accesses.

The Configuration Registers for the LPC Logical Device are located on the LPC SPB in the address range FF_3300h through FF_33F0h. The Global Configuration Registers are located within the Global Configuration Logical Device.

4.3 Basechip Logical Devices

Logical devices described in this section are peripherals that are located on the MEC1609/MEC1609i basechip and are accessible to the Host over the LPC bus.

Each logical device on the MEC1609/MEC1609i can have a set of Runtime Register and a set of Configuration Registers. The distinction between Runtime and Configuration registers is that the Host can access Runtime Registers by a direct I/O address, while it can only access Configuration Registers through a configuration port. The Embedded Controller (EC) can access all Configuration Registers and all Runtime Registers directly. The Logical Device Numbers for the Logical Devices resident in the MEC1609/MEC1609i are listed in Table 4-18, "MEC1609/MEC1609i Configuration Register Map," on page 67.

TABLE 4-1: BASECHIP LOGICAL DEVICES

Logical Device Number	Logical Device	Logical Device CR Map on Table 4-18
0h	Mailbox Interface	on page 67
1h	Keyboard Controller (8042)	on page 67
2h	ACPI EC Channel 0	on page 67
3h	ACPI EC Channel 1	on page 67
4h	ACPI EC Channel 2	on page 67
5h	ACPI EC Channel 3	on page 67
6h	ACPI PM1	on page 67
7h	UART	on page 67
8h	Legacy (Fast KB)	on page 67
Ch	LPC Interface	on page 67
Dh	VLPC Interface	on page 69
Eh	Embedded Flash	on page 70
Fh	Flash SPI	on page 70
10h	EM Interface	on page 70
3Fh	Global Configuration	on page 70

Each Companion device may have an additional 16 Logical Devices; Configuration Registers for these Logical Devices are described in the specifications for each Companion.

All Configuration and Runtime Registers in the MEC1609/MEC1609i have an assigned AHB address between FF 0000h and FF FFFFh. Configuration and Runtime Registers in Companion devices are assigned AHB addresses between FE_0000h and FE_FFFFh. Unless indicated otherwise, the EC can issue reads and writes to any register in that AHB address range. The EC can access 8-bit registers with 8-bit reads and writes, 16-bit registers with either 8-bit or 16-bit reads and writes and 32-bit registers with 8-bit, 16-bit and 32-bit reads and writes.

The Host can only access a subset of the AHB address space, and within that space all registers are treated as 8-bit registers, although a register may be implemented as a 32-bit register and accessible to the EC as a 32-bit register. The Host accesses registers in the FF_0000h through FF_FFFFh range through LPC I/O cycles. I/O cycles are mapped according to rules described in Section 4.5, on page 55, Section 4.6, on page 56 and Section 4.7, on page 60.

4.4 Registers

The Host Interface has its own Logical Device Number and Base Address as indicated in Table 4-2. The Host LPC I/O addresses for the Logical Device Configuration are selected via a Base Address Register. LPC access to configuration registers is through the Host Access Configuration Port

The Logical Device Configuration also has a Global Configuration block which has a separate Logical Device Number and Base Address Register as indicated in Table 4-2. The Base Address Register for the Global Configuration has only one writable bit, the Valid Bit, since the only I/O accessible Register has a fixed address.

Table 4-3 is a register summary for the LPC Interface block and Table 4-19, “Chip-Level (Global) Control/Configuration Registers,” on page 71 is a register summary for the Global Configuration block.

TABLE 4-2: Host Interface BASE ADDRESS TABLE

Host Interface **Blocks**	**LDN from (**Table 3-2 on page 48**)**	**AHB Base Address**
LPC Interface	Ch	FF_3000h
Global Configuration	3Fh	FF_FC00h

Note: The Host LPC I/O addresses for this instance are selected via a Base Address Register (see Section 4.6.2, on page 56). LPC access to configuration registers is through the Host Access Configuration Port (see Section 4.5.1, on page 55).

Table 4-3 is a register summary for the Host Access Port block. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port address via its LDN indicated in Table 4-2 on page 54 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 4-3: Host Access Port REGISTER SUMMARY

	Host I/O Access			**EC Interface**			
Port Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
CONFIG PORT	00h	00h	W				
INDEX PORT	00h	00h	R/W				
DATA PORT	01h	01h	R/W				

Note: The EC does not have access to the Host Access Port; however, the EC can access registers with AHB addresses.

4.5 Configuration Registers

4.5.1 HOST ACCESS PORT

The Host can access Configuration Registers through a port described in Section 4.5.2, on page 55. Host accesses are limited to 8 bits. There are 48 8-bit Global Configuration Registers (at offsets 00h through 2Fh), plus up to 208 8-bit registers associated with each Logical Device. The Logical Device is selected with the Logical Device Number Register (Global Configuration Register 07h). The INDEX PORT is used to select a specific logical device register. These registers are then accessed through the DATA PORT. The Logical Device registers are accessible only when the device is in the Configuration State.

Only two states are defined (Run and Configuration). In the Run State, the chip will always be ready to enter the Configuration State.

The desired configuration registers are accessed in two steps:

a) Write the index of the Logical Device Number Configuration Register (i.e., 07h) to the INDEX PORT and then write the number of the desired logical device to the DATA PORT

b) Write the address of the desired configuration register within the logical device to the INDEX PORT and then write or read the configuration register through the DATA PORT.

Note 1: If accessing the Global Configuration Registers, step (a) is not required.

2: Any write to an undefined or reserved Configuration register is terminated normally on the LPC bus without any modification of state in the basechip or Companion device. Any read to an undefined or reserved Configuration register returns FFh.

4.5.2 PRIMARY CONFIGURATION ADDRESS DECODER

The logical devices are configured through three Configuration Access Ports (CONFIG, INDEX and DATA). The BIOS uses these ports to initialize the logical devices at POST (Table 4-4).

The Base Address of the Configuration Access Ports is determined by the BAR that corresponds to Logical Device Ch, the LPC Interface. This is the first BAR in the table, at AHB address FF_3360h. The Configuration Access Port BAR is unique in that an LPC I/O access that matches this BAR does not directly generate an AHB read or write. Instead, the Device and Frame values in the BAR indicates that the LPC I/O should be handled locally in the LPC Logical Device. The Configuration map will issue an AHB read or write, the results of which will be used to complete the LPC access.

TABLE 4-4: MEC1609/MEC1609I CONFIGURATION ACCESS PORTS

Port Name	Relative Address	Type	Port Name
CONFIG PORT	Configuration Access Ports Base Address + 0	Write	CONFIG PORT
INDEX PORT	Configuration Access Ports Base Address + 0	Read/Write	INDEX PORT
DATA PORT	Configuration Access Ports Base Address + 1		DATA PORT

4.5.2.1 Entering the Configuration State

The INDEX and DATA ports are effective only when the chip is in the Configuration State. The device enters the Configuration State when the Config Entry Key is successfully written to the CONFIG PORT.

Config Entry Key = < 55h>

4.5.2.2 Exiting the Configuration State

The device exits the Configuration State when the following Config Exit Key is successfully written to the CONFIG PORT address.

Config Exit Key = < AAh>

4.5.2.3 Read Accessing Configuration Port

The data read from the Configuration Port is undefined when not in the Configuration State. Writing the Config Entry Key puts the chip in the Configuration State. Once in the Configuration State, reading the Configuration Port will return the last value written to the Configuration Index. If no value was written the Configuration Port reads 00h.

4.5.3 CONFIGURATION SEQUENCE EXAMPLE

To program the configuration registers, the following sequence must be followed:

1. Enter Configuration State
2. Program the Configuration Registers
3. Exit Configuration State.

The following is an example of a configuration program in Intel 8086 assembly language.

```
;----------------------------.
; ENTER CONFIGURATION STATE  |
;----------------------------'
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AX,055H ; Config Entry Key
OUT DX,AL
;----------------------------.
; CONFIGURE BASE ADDRESS,    |
; LOGICAL DEVICE 8           |
;----------------------------'
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AL,07H
OUT DX,AL ; Point to LD# Config Reg
MOV DX,CONFIG_PORT_BASE_ADDRESS+1
MOV AL, 08H
OUT DX,AL ; Point to Logical Device 8
;
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AL,60H
OUT DX,AL ; Point to BASE ADDRESS REGISTER
MOV DX,CONFIG_PORT_BASE_ADDRESS+1
MOV AL,02H
OUT DX,AL ; Update BASE ADDRESS REGISTER
;----------------------------.
; EXIT CONFIGURATION STATE   |
;----------------------------'
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AX,0AAH ; Config Exit Key
OUT DX,AL.
```

4.5.4 CONFIGURATION REGISTER ADDRESS MAPPING

The INDEX PORT defines 256 bytes for configuration. The first 48 of these bytes are Global Configuration registers, which reside in the first 48 bytes of the Configuration part of the address frame for Logical Device 3Fh. Values of INDEX greater than 48 map into registers that are specific to the Logical Device specified in the Global Configuration Logical Device Number Register 7h. These registers reside in upper 20 bytes of the Logical Device address frame. See Section 4.10.2, on page 66 for details.

4.6 Configuring Runtime Register Addresses

4.6.1 RUNTIME REGISTERS

Runtime Registers are registers that are accessible to the Host within the Host I/O address space. These Host I/O accesses are all mapped into the MEC1609/MEC1609i AHB address space onto devices located on the LPC SPB. Runtime registers all reside within the first 256 bytes of a 1KB Logical Device address frame. The Host accesses these registers with 8-bit LPC I/O accesses. Each 8-bit I/O address is mapped into an 8-bit address in the AHB address space, so the first 256 bytes of the Logical Device frame can accommodate 256 LPC Runtime Registers per Logical Device. The Host I/O addresses are determined by a block of Base Address Registers located in the LPC Logical Device. The Embedded Controller can access all the Runtime Registers as well, using loads and stores to full AHB addresses.

4.6.2 BASE ADDRESS REGISTERS

Each Logical Device has a Base Address Register (BAR), including Logical Devices that reside in the base chip as well as those that reside in Companion devices attached to the VLPC bus. These BARs are located in blocks of Configuration Registers in Logical Device 0Ch, in the AHB address range FF_3360h through FF_3384h for Logical Devices on the basechip and in the range FF_33B0h through FF_33ECh for Logical Devices on the VLPC bus, for a total of 26 BARs. On every LPC bus I/O access all Base Address Registers are checked in parallel and if any matches the LPC I/O address the MEC1609/MEC1609i claims the bus cycle.

Note: Software should that insure that no two BARs map the same LPC I/O address. If two BARs do map to the same address, the BAR_Conflict bit in the Host Bus Error Register is set when an LPC access targeting the BARConflict address. An EC interrupt can be generated.

Each BAR is 32 bits wide. The format of each BAR is summarized in Table 4-5, "Base Address Register Format". An LPC I/O request is translated by the BAR into an 8-bit read or write transaction on the AHB bus. The 16-bit LPC I/O address is translated into a 24-bit AHB address

The Base Address Register Table is itself part of the AHB address space. It resides in the Configuration quadrant of Logical Device Ch, the LPC Interface.

TABLE 4-5: BASE ADDRESS REGISTER FORMAT

BYTE3 BIT	D31	D30	D29	D28	D27	D26	D25	D24
BIT NAME	LPC Host Address, most significant bits							
BYTE2 BIT	D23	D22	D21	D20	D19	D18	D17	D16
BIT NAME	LPC Host Address, least significant bits							
BYTE1 BIT	D15	D14	D13	D12	D11	D10	D9	D8
BIT NAME	Valid	Device	Frame					
BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	Reserved	Mask						

MASK

These 7 bits are used to mask off address bits in the address match between an LPC I/O address and the Host Address field of the BARs, as described in Section 4.6.3, "Mapping LPC I/O Addresses". A block of up to 128 8-bit registers can be assigned to one base address.

FRAME

These 6 bits are used to specify a logical device frame number within a bus. This field is multiplied by 400h to provide the frame address within the peripheral bus address. If DEVICE is 0, then the logical device is located on the MEC1609/MEC1609i and all 6 bits are used for the frame number. In the MEC1609/MEC1609i Frame values for frames corresponding to logical devices that are not present on the MEC1609/MEC1609i are invalid. If DEVICE is 1, then the logical device is located on a Companion. Table 4-6, "Logical Device Bus Location" shows the relationship among DEVICE, FRAME, AHB addresses and the physical location of a logical device.

TABLE 4-6: LOGICAL DEVICE BUS LOCATION

Device	Frame	AHB Addresses	Peripheral Bus
0	00h - 3Fh	FF_0000h - FF_FFFFh	Reserved
1	00h - 0Fh	FE_0000h - FE_3FFFh	VLPC bus, Companion 0
1	10h - 1Fh	FE_4000h - FE_7FFFh	VLPC bus, Companion 1
1	20h - 2Fh	FE_8000h - FE_ BFFFh	VLPC bus, Companion 2
1	30h - 3Fh	FE_C000h - FE_FFFFh	VLPC bus, Global Registers

DEVICE

This bit combined with FRAME constitute the Logical Device Number. DEVICE identifies the physical location of the logical device. If this bit is 0, the logical device is located on the MEC1609/MEC1609i and the AHB address is on the local AHB bus. If this bit is 1, the logical device is located on a Companion.

VALID

If this bit is 1, the BAR is valid and will participate in LPC matches. If it is 0 this BAR is ignored

HOST_ADDRESS

These 16 bits are used to match LPC I/O addresses

4.6.3 MAPPING LPC I/O ADDRESSES

A Base Address Register will match an LPC I/O address, and thus the MEC1609/MEC1609i will claim the LPC bus cycle, if the following relation holds:

(LPC Address & ~BAR.MASK) == (BAR.LPC_Address & ~BAR.MASK) && (BAR.Valid == 1)

If one of the BARs match, the LPC cycle will be claimed and the LPC request will be translated to an AHB address according to the following formulae:

DEVICE = 0:

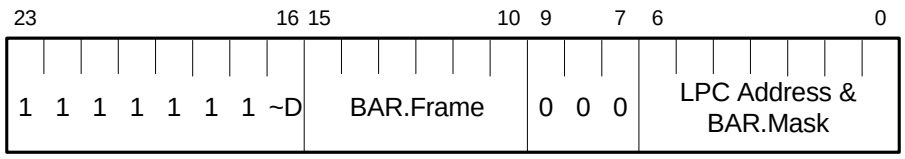
AHB Address = FF_000h | (BAR.Frame<<10) | (LPC_Address & BAR.MASK)

DEVICE = 1:

AHB Address = FE_000h | (BAR.Frame<<10) | (LPC_Address & BAR.MASK)

The formulae are illustrated in Figure 4-1, "LPC BAR Mapping":

FIGURE 4-1: LPC BAR MAPPING



~D = ~BAR.Device

When matching LPC I/O addresses, the MEC1609/MEC1609i ignores address bits that correspond to ‘1b’ bits in the MASK field. When forming the AHB address from the LPC I/O address, the LPC I/O address bits that correspond to the ‘1b’ bits in the MASK are passed through to the AHB address. For example, the Keyboard Controller (9042 Interface) Base Address Register has 60h in the LPC Address field, the Frame field is 01h, and the MASK field is 04h. Because of the single ‘1b’ bit in MASK, the BAR will match LPC I/O patterns in the form ‘00000000011000hb’, so both 60h and 64h will be matched and claimed by the MEC1609/MEC1609i. When forming the AHB address, the Frame number 01h will be put in bit positions 15 through 10 and concatenated with FF_0000h, to form the Keyboard Controller Frame base address of FF_0400h. If the Host reads from address 60h, a read access will be initiated to AHB address FF_0400h (the base address OR’d with bit 2 from 60h). If the Host reads from address 64h, a read access will be initiated to AHB address FF_0404h (again, bit 2 from the LPC I/O address is OR’d into the AHB address).

As another example, if a standard 16550 UART was located at LPC I/O address 238h, then the UART Receive buffer would appear at address 238h and the Line Status register at 23Dh. If the BAR for the UART was set to 0238_8047h, then the UART will be matched at I/O address 238h, the UART is located on the basechip in Logical Device 6h, the UART is enabled and the UART device includes 8 registers. The Receive buffer would map to AHB address FF_1800h and the Line Status register would map to AHB address FF_1805h.

4.6.4 BASE ADDRESS REGISTER TABLE

Table 4-7, "Base Address Registers Default Values", lists the Base Address Registers for all logical devices on the MEC1609/MEC1609i base chip. The BAR EC Offsets are relative to the base address of the LPC Logical Device, which is located at AHB address FF_3000h. The columns to the right of the heavy line show the field definitions for the default values listed in the column labeled “Reset Default”. The 16 BARs at offsets 3B0h-3ECh are available for logical devices on the VLPC bus. Shaded fields in Table 4-7 are read-only. Because the DEVICE field is always 0 for BAR registers at offsets 360h-38Ch, these BARs are restrained to logical devices on the basechip. Because the DEVICE field is always 1 for BAR registers at offsets 3B0h-3ECh, these 16 BARs are restrained to logical devices on Companions.

The EC can read and write the BAR table entries with 32-bit, 16-bit or 8-bit accesses. The Host accesses the BAR table using 8-bit reads and writes.

Note: The BAR at offset 360h associated with the LPC Interface is different in that LPC I/O accesses that are claimed by this BAR does not translate directly into a AHB addresses. Addresses that are claimed by this BAR, the Configuration Port BAR, are used to manage Configuration Registers. These accesses are described in Section 4.5, on page 55. This BAR is also special in that a byte write to offset 362h (bits[7:0] of the LPC Host Address field) does not directly write into the BAR. Instead, the byte is held in a buffer. A byte write to offset 363h will both write the byte at offset 363h and will copy the buffer into offset 362h. This is done to insure that the 16-bit LPC Host Address field of the Configuration Port BAR is always completely updated in one cycle.

The shaded LPC I/O Address, VALID, DEVICE, FRAME, MASK fields are read-only Table 4-7. The unshaded fields has read/write access.

TABLE 4-7: BASE ADDRESS REGISTERS DEFAULT VALUES

Bar EC Offset	LPC Offset	Result Default	LPC I/O Address	Valid	Device	Frame	Mask	Description
360h	60h	002E_0C01h	002Eh	0	0	C	1	Logical Device 0Ch: LPC Interface (Configuration Port)
364h	64h	0000_0001h	0000h	0	0	0	1	Logical Device 00h: Mailbox Register I/F
368h	68h	0060_0104h	0000h	0	0	1	4	Logical Device 01h: Keyboard Controller (8042 Interface)
36Ch	6Ch	0062_0204h	0062h	0	0	2	4	Logical Device 02h: ACPI EC Interface 0
370h	70h	0062_0307h	0062h	0	0	3	7	Logical Device 03h: ACPI EC Interface 1
374h	74h	0062_0407h	0062h	0	0	4	7	Logical Device 04h: ACPI EC Interface 2
378h	78h	0062_0507h	0062h	0	0	5	7	Logical Device 05h: ACPI EC Interface 3
37Ch	7Ch	0000_0607h	0000h	0	0	6	7	Logical Device 06h: ACPI PM1 Interface
380h	80h	0000_0707h	0000h	0	0	7	7	Logical Device 07h: UART
384h	84h	0092_0800h	0092h	0	0	8	0	Logical Device 08h: Legacy (GATEA20) I/F
388h	88h	0000_0E04h	0000h	0	0	E	4	Logical Device 0Eh: Embedded Flash Interface
38Ch	8Ch	0000_0F07h	0000h	0	0	F	07h	Logical Device 0Fh: GPSPI for Flash Interface
390h	90h	0000_100Fh	0000h	0	0	10h	Fh	Logical Device 10h. EM Interface
3B0h-3ECh	B0h - ECh	0000_4000h	0000h	0	1	0	0	BAR available for VLPC Logical Devices

Note 4-1 In order to avoid address conflict with the Keyboard Controller (8042) LDN 1h at legacy at LPC I/O address 60h/64h, only ACPI EC Channel 0 LDN 2h at AHB base address FF_0800h should be located at the legacy LPC I/O address 62h/66h. When operating both the Keyboard Controller (8042)

and ACPI EC Channel 0 at legacy LPC I/O addresses 60h/62h/64h/66h, the Mask value in ACPI EC Channel 0 BAR should be programmed to 4h and the ACPI EC Channel 0, EC_Cx_Byte Count Register, Byte Count Register on page 166 should remain 0. To utilize the ACPI EC Channel 0 in Non Legacy Operation the ACPI EC Channel 0 BAR must be placed on LPC I/O 8 byte boundary. (See Section 9.5, on page 166). Only bits[3:0] in the Mask field of ACPI EC Channel 0 BAR are programmable, bits[6:4] are read-only '000'.

Note 4-2 The ACPI EC Channel 1, ACPI EC Channel 2 and ACPI EC Channel 3 (LDN 3h, 4h and 5h) can not be located at the legacy LPC I/O address 62h/66h. The ACPI EC Channel 1, ACPI EC Channel 2 and ACPI EC Channel 3 BAR's must be placed on LPC I/O 8 byte boundaries.

4.7 DMA

LPC DMA cycles are mapped to AHB memory addresses by the LPC Logical Device. The addresses reference FIFOs that are associated with the DMA, and can be located either on the basechip or in a Companion device attached to the VLPC bus. FIFO addresses are restricted to 32-bit aligned addresses (that is, addresses that are divisible by 4). Configuration for DMA Device access is described in Section 4.7.1, on page 60.

4.7.1 DMA CONFIGURATION REGISTERS

The MEC1609/MEC1609i will claim an LPC DMA request if the requested channel is listed as valid in the Table 4-8, "DMA Configuration Register Map". A channel is claimed if the DMA Configuration Register Format that corresponds to the channel is maps to a Logical Device. In order to execute the DMA operation, the MEC1609/MEC1609i translates the DMA access into a bus read or write of the FIFO that corresponds to channel in question. The address of a DMA FIFO will always be one of the first 16 32-aligned addresses within the DMA quadrant of a Logical Device frame.

The AHB address can be on either the basechip or in one of the Companion Devices on the VLPC bus.The 15 VLPC events (5 events from each of the 3 possible VLPC Companion devices) can all be mapped to any of the DMA channels.

The mapping in the DMA Configuration Register Map is used both for mapping LPC DMA I/O requests from the Host to Logical Devices, as well as for mapping DMA requests from Logical Devices to the LPC Bus LDRQ# DMA request signal.

The Host can access the DMA Configuration registers with 8-bit accesses. The EC can access the DMA Configuration registers as four 32-bit registers, eight 16-bit registers or sixteen 8-bit registers.

The mapping of DMA devices into the AHB address space is shown in Figure 4-2, "DMA Address Mapping":

FIGURE 4-2: DMA ADDRESS MAPPING

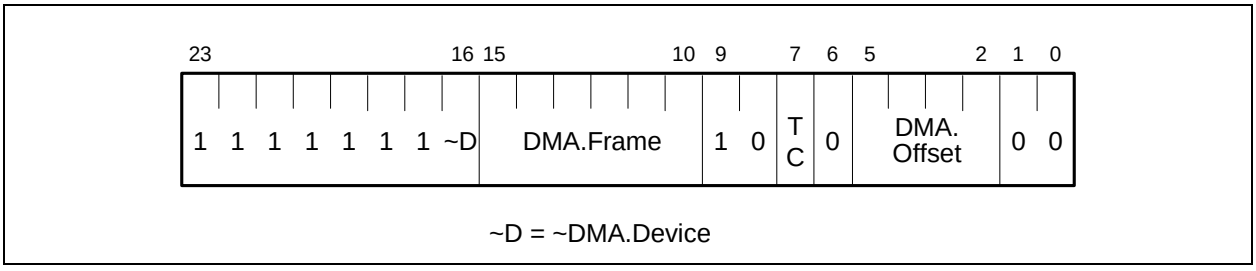


TABLE 4-8: DMA CONFIGURATION REGISTER MAP

Address	Type	Reset	Configuration Register Name
FF_3350h	R/W	0000h	DMA Channel 0
FF_3352h	R/W	0000h	DMA Channel 1
FF_3354h	R/W	0000h	DMA Channel 2
FF_3356h	R/W	0000h	DMA Channel 3
FF_3358h	R	0000h	DMA Channel 4 (Reserved)
FF_335Ah	R/W	0000h	DMA Channel 5
FF_335Ch	R/W	0000h	DMA Channel 6
FF_335Eh	R/W	0000h	DMA Channel 7

Note 4-3 DMA Channel 4 is reserved in the MEC1609/MEC1609i. LPC Host cycles with DMA channel 4 asserted will be unclaimed by the MEC1609/MEC1609i.

TABLE 4-9: DMA CONFIGURATION REGISTER FORMAT

BYTE1 BIT	D15	D14	D13	D12	D11	D10	D9	D8
BIT NAME	Valid	Device	Frame					
BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	Event				Offset			

OFFSET

These 4 bits select which register within the Logical Device address space are to be used for this DMA. The 4 addresses correspond to the following address offsets:

0h: +200h
1h: +204h
2h: +208h
3h: +20Ch
4h: +210h
5h: +214h
6h: +218h
7h: +21Ch
8h: +220h
9h: +224h.
Ah: +228h
Bh: +22Ch
Ch: +230h
Dh: +234h
Eh: +238h
Fh: +23Ch

EVENT

If DEVICE is 1, the DMA Request for this channel is generated by a VLPC Bus Companion identified by Bits[13:12] from Table 4-10, "DMA Request Logical Device Selection". If this field has a value Fh no DMA Request is generated.

If DEVICE is 0, the DMA Request for this channel is sourced from the MEC1609/MEC1609i Logical Device identified by FRAME and this field is ignored.

TABLE 4-10: DMA REQUEST LOGICAL DEVICE SELECTION

Device	Event	DMA Request Source
0	0h - Fh	DMA Request on basechip; the EVENT field is ignored
1	00h - 04h	Event 0 through Event 4 from VLPC Companion 0
1	05h - 09h	Event 0 through Event 4 from VLPC Companion 1
1	0Ah - 0Eh	Event 0 through Event 4 from VLPC Companion 2
1	Fh	Reserved

FRAME

These 6 bits are used to a logical device frame number within a bus. This field is multiplied by 400h to provide the frame address within the peripheral bus address. If DEVICE is 0, then the logical device is located on the MEC1609/MEC1609i and all 6 bits are used for the frame number. In the MEC1609/MEC1609i Frame values greater than 0Ah are invalid, since there are only 11 logical devices accessible to the Host. If DEVICE is 1, then the logical device is located on a Companion, and the 6 bit frame number is limited to values between 00h and 2F.h Table 4-11, "Logical Device Bus Location" shows the relationship among DEVICE, FRAME, AHB addresses and the physical location of a logical device.

TABLE 4-11: LOGICAL DEVICE BUS LOCATION

Device	Frame	AHB Addresses	Peripheral Bus
0	00h - 3Fh	FF_0000h - FF_FFFFh	Reserved
1	00h - 0Fh	FE_0000h - FE_3FFFh	VLPC bus, Companion 0
1	10h - 1Fh	FE_4000h - FE_7FFFh	VLPC bus, Companion 1
1	20h - 2Fh	FE_8000h - FE_BFFFh	VLPC bus, Companion 2
1	30h - 3Fh	Reserved	Reserved

DEVICE

This bit combined with FRAME constitute the Logical Device Number. DEVICE identifies the physical location of the logical device. If this bit is 0, the logical device is located on the MEC1609/MEC1609i and the AHB address is on the local AHB bus.If this bit is 1, the logical device is located on a Companion. Logical devices with Logical Device Numbers between 00h and 3Fh are located on the basechip and logical devices with Logical Device Numbers between 40h and 6Fh are located on Companions.

VALID

If this bit is 1, the DMA Channel is active on the MEC1609/MEC1609i or on a VLPC Companion and will be claimed by the MEC1609/MEC1609i. If it is 0 this DMA Channel is ignored.

4.8 SERIRQ Interrupts

The MEC1609/MEC1609i can routes Logical Device interrupts onto SIRQ stream frames IRQ[0:15]. Routing is controlled by the SIRQ Interrupt Configuration Registers. There is one SIRQ Interrupt Configuration Register for each accessible SIRQ Frame (IRQ); all 16 registers are listed in Table 4-12, "SIRQ Interrupt Configuration Register Map". Each SIRQ Interrupt Configuration Register controls a series of multiplexors which route to a single Logical Device interrupt as illustrated in FIGURE 4-3: SIRQ Routing Internal & VLPC Logical Devices on page 65. The format for each SIRQ Interrupt Configuration Register is described in Table 4-13. Each Logical Device can have up to two LPC SERIRQ interrupts. When the MEC1609/MEC1609i is polled by the host, each SIRQ frame routes the level of the Logical Device interrupt (selected by the corresponding SIRQ Interrupt Configuration Register) to the SIRQ stream.

Note: Two Logical Devices cannot share a Serial IRQ.

The SIRQ Interrupt Configuration Register The Host can access the Interrupt Configuration registers with 8-bit accesses. The EC can access the Interrupt Configuration registers as four 32-bit registers, eight 16-bit registers or sixteen 8-bit registers.

Note: An interrupt is deactivated by setting an entry in the SIRQ Interrupt Configuration Register Map to FFh, which is the default reset value.

4.8.1 VLPC BUS DEVICE INTERRUPTS

An Event from a VLPC Bus Companion can be mapped to both a SERIRQ LPC interrupt, by the SIRQ Interrupt Configuration Register Map, as well as to an EC interrupt.

A TPM located on the VLPC Bus will return an interrupt signal and a 4-bit Serial IRQ channel number to the LPC interface on the MEC1609/MEC1609i. This interrupt is merged with the SERIRQ signals generated by the SIRQ Interrupt Configuration Register Map when the SERIRQ poll is generated. A TPM Serial IRQ channel of 0 implies that TPM interrupts are disabled, so the TPM interrupt is not merged with $SERIRQ_0$. A TPM Serial IRQ channel of *i*, where *i* is from 1 to 15, overrides the definition of SERIRQ for channel *i* in the SIRQ Interrupt Configuration Register Map, and the MEC1609/MEC1609i will claim $SERIRQ_i$ even if the map contains FFh, the disable code. The TPM SERIRQ channel cannot be shared with any other device. See FIGURE 4-3: SIRQ Routing Internal & VLPC Logical Devices on page 65.

4.8.2 SERIRQ CONFIGURATION REGISTERS

TABLE 4-12: SIRQ INTERRUPT CONFIGURATION REGISTER MAP

Address	Type	Reset	Configuration Register Name
FF_3340h	R/W	FFh	IRQ0
FF_3341h	R/W	FFh	IRQ1
FF_3342h	R/W	FFh	IRQ2 (nSMI)
FF_3343h	R/W	FFh	IRQ3
FF_3344h	R/W	FFh	IRQ4
FF_3345h	R/W	FFh	IRQ5
FF_3346h	R/W	FFh	IRQ6
FF_3347h	R/W	FFh	IRQ7
FF_3348h	R/W	FFh	IRQ8
FF_3349h	R/W	FFh	IRQ9
FF_334Ah	R/W	FFh	IRQ10
FF_334Bh	R/W	FFh	IRQ11
FF_334Ch	R/W	FFh	IRQ12
FF_334Dh	R/W	FFh	IRQ13
FF_334Eh	R/W	FFh	IRQ14
FF_334Fh	R/W	FFh	IRQ15

Note 4-4 The SIRQ Interrupt Configuration Registers are through the Host Access Port as 8-bit accesses. The EC can access the SIRQ Interrupt Configuration Registers as 32-bit, 16-bit across 8-bit boundary or as individual 8-bit accesses.

TABLE 4-13: SIRQ INTERRUPT CONFIGURATION REGISTER FORMAT

BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	Select	Device	Frame					

FRAME

When combined with DEVICE, these six bits select the Logical Device for on-chip devices as the source for the interrupt. For VLPC Companions, this field, combined with DEVICE, selects the Event bit to be mapped to the SIRQ, according to Table 4-14, "Interrupt Logical Device Selection".

This field defaults to 3Fh.

TABLE 4-14: INTERRUPT LOGICAL DEVICE SELECTION

Device	Frame	Interrupt Source
0	00h - 3Fh	Logical Devices 00h through 3Fh on LPC SPB on the MEC1609/MEC1609i
1	00h - 04h	Event 0 through Event 4 from VLPC Companion 0
1	05h - 09h	Event 0 through Event 4 from VLPC Companion 1
1	0Ah - 0Eh	Event 0 through Event 4 from VLPC Companion 2
1	0Fh - 3Fh	Interrupt source is not in the MEC1609/MEC1609i or any VLPC Companion

Note: The LPC Logical Device (Logical Device Number 0Ch) can be used by the Embedded Controller to generate a Serial Interrupt Request to the Host under software control.

DEVICE

This bit combined with FRAME constitute the Logical Device Number of the interrupt source. DEVICE identifies the physical location of the logical device. If this bit is 0, the logical device is located on the MEC1609/MEC1609i. If this bit is 1, the logical device is located on a Companion and the interrupt source is on of the five Events that are transmitted from the VLPC Bus Companion devices.

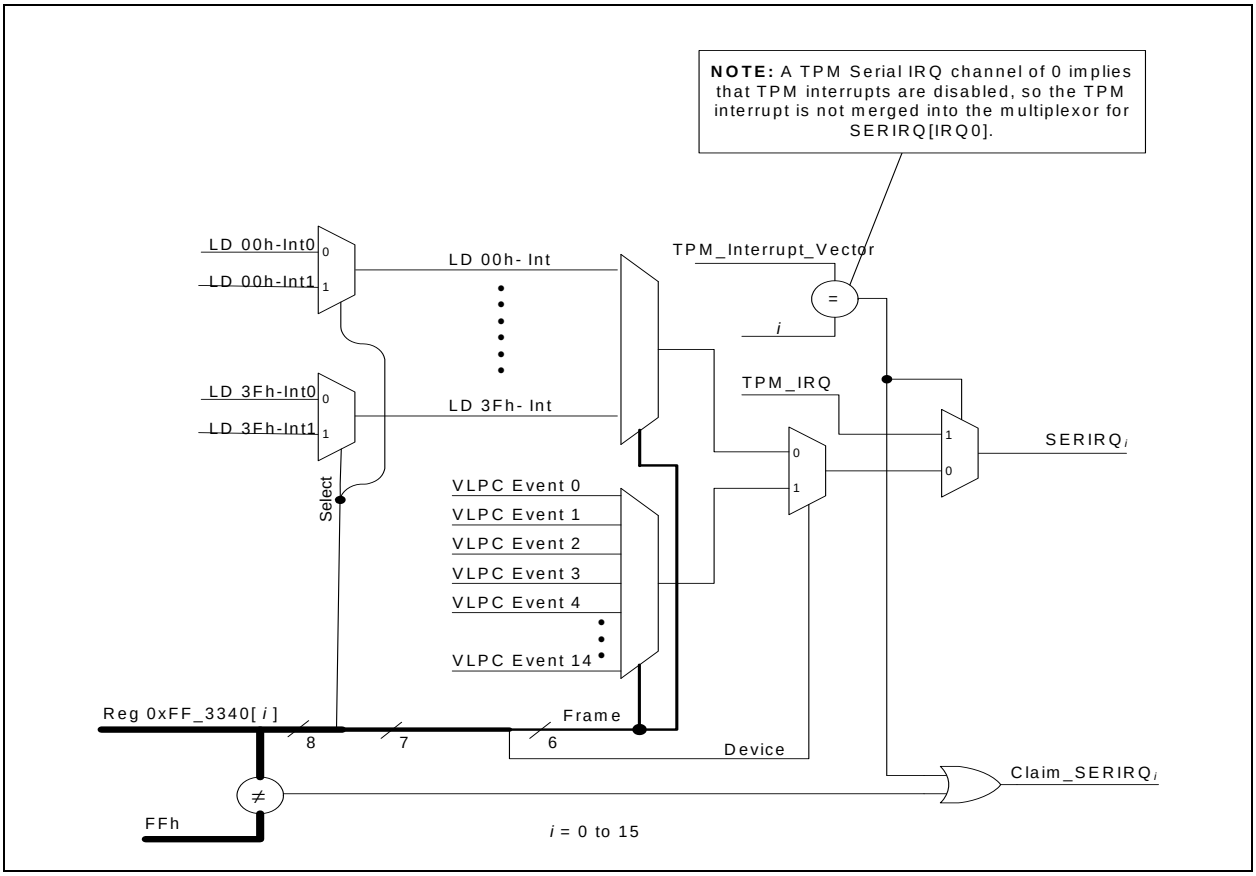
This field defaults to 1.

SELECT

If this bit is 0, the first interrupt signal from the Logical Device is selected for the SERIRQ vector. If this bit is 1, the second interrupt signal from the Logical Device is selected. The KYBD controller is the only Logical Devices on the MEC1609/MEC1609i which has a second interrupt signal. For all other Logical Devices this field is ignored.

This field defaults to 1.

FIGURE 4-3: SIRQ ROUTING INTERNAL & VLPC LOGICAL DEVICES



4.8.2.1 MEC1609/MEC1609i SIRQ Routing

TABLE 4-15: MEC1609/MEC1609I LOGICAL DEVICE SIRQ ROUTING

SIRQ Interrupt Configuration Register			Logical Device Interrupt Source
SELECT	DEVICE	FRAME	
0	0	00h	MailBox SIRQ - See Section 12.3.1, on page 194
1	0	00h	SMI - Section 12.3.1, on page 194
0	0	01h	Keyboard SIRQ - Section 10.4.1, on page 171
1	0	01h	Mouse SIRQ - Section 10.4.1, on page 171
0	0	06h	UART SIRQ - Section 13.3.2, on page 203
0	0	0Ch	Serial IRQ - Section 6.10.3, on page 129
1	1	X	VLPC Companion Logical Device - See Table 4-14 & VLPC companion specification

4.9 Configuration Register Reset Conditions

There are two reset conditions that will cause Configuration Registers on the MEC1609/MEC1609i to reset to default values. A reset can be caused by a VTR Power On Reset condition or a VCC-Powergood Power on Reset condition. In addition, firmware running on the Embedded Controller can set all Configuration Registers to a default condition. The Host can request that the Configuration Registers be reset through a request to the Embedded Controller sent via the Mailbox interface.

Logical Devices resident on Companion devices are reset independently. Please refer to the VLPC specification.

4.10 Logical Device Configuration/Control Registers

A separate set of control and configuration registers exist for each Logical Device and is selected with the Logical Device # Register (07h). The Logical Devices are listed in Table 4-1, “Basechip Logical Devices,” on page 53, and the registers within each Logical Device are listed in Section 4.10.2, on page 66.

4.10.1 LOGICAL DEVICE ACTIVATION

Many Logical Devices have a register, called Activate, that is used to activate the Logical Device. When a Logical Device is inactive, it is powered down and will not respond to an AHB request. The format for the Activate Register is shown in Table 4-16, "Activate Register". The Activate Register for the LPC Logical Device is shown in Table 6-6, “Activate Register,” on page 126.

TABLE 4-16: ACTIVATE REGISTER

HOST OFFSET	BYTE0: 30h			8-bit			**HOST SIZE**	
EC OFFSET	Frame offset 330h			8-bit			**EC SIZE**	
POWER	VTR			00b			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the logical device is powered and functional. When this bit is 0, the logical device is powered down and inactive.

4.10.2 CONFIGURATION REGISTER MAP

The MEC1609/MEC1609i Configuration register map is shown in Table 4-18, "MEC1609/MEC1609i Configuration Register Map". Logical Device numbers are in hexadecimal. All Logical Devices are accessible by both the Host and the EC. Logical Devices between 00h and 3Fh are located on the basechip. Logical Devices between 40h and 4Fh are located on the VLPC Bus Companion 0, Logical Devices between 50h and 5Fh are located on the VLPC Bus Companion 1 and Logical Devices between 60h and 6Fh are located on the VLPC Bus Companion 2. Logical Devices between 70h and 7Fh are used to access addresses in the VLPC Global Register space.

The EC has access to all Configuration Registers and all Global Control registers directly on the AHB bus. The address of a Global Control register is FF FF00h+*offset*, where *offset* is the offset listed in Column 2 of Table 4-19, "Chip-Level (Global) Control/Configuration Registers". The other Configuration Registers are accessible at the INDEX (Table 4-18, Column 1) plus the AHB Address Base (Table 4-17, "Configuration Register AHB Mapping", Column 2). LDN is the Logical Device Number.

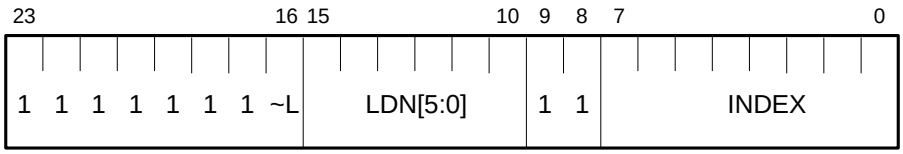
Note: The Global Configuration Registers are the first 48 bytes of the configuration space Logical Device 3Fh.

TABLE 4-17: CONFIGURATION REGISTER AHB MAPPING

Logical Device Range	AHB Address Base
00h - 3Fh	FF_0300h + (LDN << 10)
40h - 7Fh	FE_0300h + ((LDN - 40h) << 10)

The Configuration Register address mapping is illustrated in:

FIGURE 4-4: CONFIGURATION REGISTER MAPPING



~L = ~LDN[6]

TABLE 4-18: MEC1609/MEC1609I CONFIGURATION REGISTER MAP

LPC CR INDEX	AHB Offset	Type Note 4-5	Reset Note 4-6	Configuration Register Name
Configuration Registers for LDN 0h (Mailbox Interface) at AHB base address FF_0000h				
-	-	-	-	None
Configuration Registers for LDN 1h (Keyboard Controller (8042)) at AHB base address FF_0400h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
Configuration Registers for LDN 2h (ACPI EC Channel 0) at AHB base address FF_0800h				
-	-	-	-	None
Configuration Registers for LDN 3h (ACPI EC Channel 1) at AHB base address FF_0C00h				
-	-	-	-	None
Configuration Registers for LDN 4h (ACPI EC Channel 2) at AHB base address FF_1000h				
-	-	-	-	None
Configuration Registers for LDN 5h (ACPI EC Channel 3) at AHB base address FF_1400h				
-	-	-	-	None
Configuration Registers for LDN 6h (ACPI PM1) at AHB base address FF_1800h				
-	-	-	-	None
Configuration Registers for LDN 7h (UART) at AHB base address FF_1C00h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
F0h	3F0h	R/W	00h on nSYS_RST	Configuration Select Register
Configuration Registers for LDN 8h (Legacy (Fast KB)) at AHB base address FF_2000h				
30h	330h	R/W	00h on nSYS_RST	PORT92 Enable Register
Configuration Registers for LDN Ch (LPC Interface) at AHB base address FF_3000h				
30h	330h	R/W	00h on nSYS_RST	Activate Register

TABLE 4-18: MEC1609/MEC1609I CONFIGURATION REGISTER MAP (CONTINUED)

LPC CR INDEX	AHB Offset	Type Note 4-5	Reset Note 4-6	Configuration Register Name
40h	340h	R/W	FFh on nSIO_RESET	SIRQ IRQ0 Configuration Register
41h	341h	R/W	FFh on nSIO_RESET	SIRQ IRQ1 Configuration Register
42h	342h	R/W	FFh on nSIO_RESET	SIRQ IRQ2 (nSMI) Configuration Register
43h	343h	R/W	FFh on nSIO_RESET	SIRQ IRQ3 Configuration Register
44h	344h	R/W	FFh on nSIO_RESET	SIRQ IRQ4 Configuration Register
45h	345h	R/W	FFh on nSIO_RESET	SIRQ IRQ5 Configuration Register
46h	346h	R/W	FFh on nSIO_RESET	SIRQ IRQ6 Configuration Register
47h	347h	R/W	FFh on nSIO_RESET	SIRQ IRQ7 Configuration Register
48h	348h	R/W	FFh on nSIO_RESET	SIRQ IRQ8 Configuration Register
49h	349h	R/W	FFh on nSIO_RESET	SIRQ IRQ9 Configuration Register
4Ah	34Ah	R/W	FFh on nSIO_RESET	SIRQ IRQ10 Configuration Register
4Bh	34Bh	R/W	FFh on nSIO_RESET	SIRQ IRQ11 Configuration Register
4Ch	34Ch	R/W	FFh on nSIO_RESET	SIRQ IRQ12 Configuration Register
4Dh	34Dh	R/W	FFh on nSIO_RESET	SIRQ IRQ13 Configuration Register
4Eh	34Eh	R/W	FFh on nSIO_RESET	SIRQ IRQ14 Configuration Register
4Fh	34Fh	R/W	FFh on nSIO_RESET	SIRQ IRQ15 Configuration Register
50h	350h	R/W	00h on nSIO_RESET	DMA Channel 0, LSB Configuration Register
51h	351h	R/W	00h on nSIO_RESET	DMA Channel 0, MSB Configuration Register
52h	352h	R/W	00h on nSIO_RESET	DMA Channel 1, LSB Configuration Register
53h	353h	R/W	00h on nSIO_RESET	DMA Channel 1, MSB Configuration Register
54h	354h	R/W	00h on nSIO_RESET	DMA Channel 2, LSB Configuration Register
55h	355h	R/W	00h on nSIO_RESET	DMA Channel 2, MSB Configuration Register
56h	356h	R/W	00h on nSIO_RESET	DMA Channel 3, LSB Configuration Register
57h	357h	R/W	00h on nSIO_RESET	DMA Channel 3, MSB Configuration Register
58h	358h	R/W	00h on nSIO_RESET	DMA Channel 4 (Reserved), LSB Configuration Register
59h	359h	R/W	00h on nSIO_RESET	DMA Channel 4 (Reserved), MSB Configuration Register
5Ah	35Ah	R/W	00h on nSIO_RESET	DMA Channel 5, LSB Configuration Register
5Bh	35Bh	R/W	00h on nSIO_RESET	DMA Channel 5, MSB Configuration Register
5Ch	35Ch	R/W	00h on nSIO_RESET	DMA Channel 6, LSB Configuration Register
5Dh	35Dh	R/W	00h on nSIO_RESET	DMA Channel 6, MSB Configuration Register
5Eh	35Eh	R/W	00h on nSIO_RESET	DMA Channel 7, LSB Configuration Register
5Fh	35Fh	R/W	00h on nSIO_RESET	DMA Channel 7, MSB Configuration Register
60h - 63h	3360h	R/W / R	002E_0C01h on nSIO_RESET	BAR for Configuration Port
64h - 67h	3364h	R/W / R	0000_0001h on nSIO_RESET	BAR for Mailbox
68h - 6Bh	3368h	R/W / R	0060_0104h on nSIO_RESET	BAR for 8042/Keyboard Interface
6C - 6F	336Ch	R/W / R	0062_0204h on nSIO_RESET	BAR for ACPI EC Interface 0 (See Note 4-1 on page 59)
70h - 73h	3370h	R/W / R	0062_0307h on nSIO_RESET	BAR for ACPI EC Interface 1 (See Note 4-2 on page 60)
74h - 77h	3374h	R/W / R	0062_0407h on nSIO_RESET	BAR for ACPI EC Interface 2 (See Note 4-2 on page 60)

TABLE 4-18: MEC1609/MEC1609I CONFIGURATION REGISTER MAP (CONTINUED)

LPC CR INDEX	AHB Offset	Type Note 4-5	Reset Note 4-6	Configuration Register Name
78h - 7Bh	3378h	R/W / R	0062_0507h on nSIO_RESET	BAR for ACPI EC Interface 3 (See Note 4-2 on page 60)
7Ch - 7Fh	337Ch	R/W / R	0000_0607h on nSIO_RESET	BAR for ACPI PM1 Interface
80h - 83h	3380h	R/W / R	0000_0707h on nSIO_RESET	BAR for UART
84h - 87h	3384h	R/W / R	0092_0800h on nSIO_RESET	BAR for Legacy (Fast KYBD) Interface
88h - 8Bh	3388h	R/W / R	0000_0E04h on nSIO_RESET	BAR for Embedded Flash Interface
8Ch - 8Fh	338Ch	R/W / R	0000_0F07h on nSIO_RESET	BAR for GP-SPI Interface
90h - 93h	3390h	R/W / R	0000_100F on nSIO_RESET	BAR for EM Interface
B0h - B3h	33B0h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
B4h - B7h	33B4h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
B8h - BBh	33B8h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
BCh - BFh	33BCh	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
C0h - C3h	33C0h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
C4h - C7h	33C4h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
C8h - CBh	33C8h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
CCh - CFh	33CCh	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
D0h - D3h	33D0h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
D4h - D7h	33D4h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
D8h - DBh	33D8h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
DCh - DFh	33DCh	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
E0h - E3h	33E0h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
E4h - E7h	33E4h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
E8h - EBh	33E8h	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
ECh - EFh	33ECh	R/W	0000_4000h on nSIO_RESET	BAR for VLPC Companion device
Configuration Registers for LDN Dh (VLPC Interface) at AHB base address FF_3400h				
30h	3730h	R/W	00h on nSIO_RESET	Activate
F0h, F1h	37F0h	R/W	0000_0000h on nSIO_RESET	Interface Control

TABLE 4-18: MEC1609/MEC1609I CONFIGURATION REGISTER MAP (CONTINUED)

LPC CR INDEX	AHB Offset	Type Note 4-5	Reset Note 4-6	Configuration Register Name
F4h, F5h	37F4h	R/WC	0000_0000h on nSIO_RESET	Error Address
F8h	37F8h	R/W	0000_0000h on nSIO_RESET	VLPC Clock Divider
Configuration Registers for LDN Eh (Embedded Flash) at AHB base address FF_3800h				
-	-	-	-	None
Configuration Registers for LDN Fh (Flash SPI) at AHB base address FF_3C00h				
-	-	-	-	None
Configuration Registers for LDN 10h (EM Interface) at AHB base address FF_4000h				
-	-	-	-	None
Configuration Registers for LDN 3Fh (Global Configuration) at AHB base address FF_FC00h				
00h - 02h	FF00h - FF02h	-		Reserved
04h - 06h	FF04h - FF06h	-	00h on nSIO_RESET	Reserved
07h	FF07h	R/W	-	Logical Device Number
08h - 1Fh	FF08h- FF1Fh	-		Reserved
20h	FF20h	R	48h hardwired	Device ID
21h	FF21h	R	Current Revision hardwired	Device Revision A read-only register which provides device revision information
22h– 23h	FF22h- FF23h	-	04h on nSIO_RESET	MCHP Reserved
24h	FF24h	R/W	00h	Device Mode
25h – 2Fh	FF25h- FF2Fh	-		MCHP Reserved

Note 4-5 R/W / R means that some parts of a register are read/write and some parts are read-only.

Note 4-6 Resets are defined in Section 5.0, "Power, Clocks and Resets": nSYS_RST on page 98 and nSIO_RESET on page 75.

4.11 Chip-Level (Global) Control/Configuration Registers [00h - 2Fh]

The chip-level (global) registers reside in Logical Device 3Fh at AHB addresses FF_FF00h through FF_FF2Fh. All unimplemented registers and bits ignore writes and return zero when read. The global registers are accessed in the configuration address range [00h - 2Fh] in all Logical Devices. There is no Activate associated with Logical Device 3Fh: the Global Configuration Registers are always accessible.

As with all Configuration Registers, the INDEX PORT is used to select a Global Configuration Register in the chip. The DATA PORT is then used to access the selected register.

The Host can access all the Global Configuration registers at the offsets listed in Table 4-19, "Chip-Level (Global) Control/Configuration Registers" through the INDEX PORT and the DATA PORT. The EC can access all these registers at the listed offsets from the AHB Base Address shown in Table 4-18, "MEC1609/MEC1609i Configuration Register Map".

TABLE 4-19: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS

Register	Offset	Description
CHIP (GLOBAL) CONTROL REGISTERS		
Reserved	00h-03h	Reserved, Writes are ignored, reads return 0.
Reserved	04h - 06h	Reserved - Writes are ignored, reads return 0.
Logical Device Number	07h	A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device. **Note:** The Activate command operates only on the selected logical device.
Reserved	08h - 1Fh	Reserved - Writes are ignored, reads return 0.
Device ID Hard Wired	20h	A read-only register which provides device identification: Bits[7:0] = 48h
Device Revision Hard Wired	21h	A read-only register which provides device revision information. Bits[7:0] = current revision when read
Reserved	22h - 23h	Reserved - Writes are ignored, reads return 0.
Device Mode	24h	Bit [1:0] Reserved – writes ignored, reads return “0”. Bit[2] SerIRQ Mode (Note 4-7) = 0: Serial IRQ Disabled. = 1: Serial IRQ Enabled (Default). Bit [7:3] Reserved – writes ignored, reads return “0”.
Reserved	25h - 27h	Reserved - Writes are ignored, reads return 0.
Test Register	28h	MCHP Test Mode Register, Reserved for Microchip
Test Register	29h	MCHP Test Mode Register, Reserved for Microchip
Reserved	2Ah - 2Bh	Reserved - Writes are ignored, reads return 0.
Test Register	2Ch	MCHP Test Mode Register, Reserved for Microchip
Clock Tree Control 0	2Dh	Bit [0] Force EC CLOCK TREE 0 on. Bit [1] Force EC CLOCK TREE 0 off. Bit [2] Force EC CLOCK TREE 1 on. Bit [3] Force EC CLOCK TREE 1 off. Bit [4] Force EC CLOCK TREE 2 on. Bit [5] Force EC CLOCK TREE 2 off. Bit [6] Force EC CLOCK TREE 3 on. Bit [7] Force EC CLOCK TREE 3 off.

TABLE 4-19: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS (CONTINUED)

Register	Offset	Description
Clock Tree Control 1	2Eh	Bit [0] Force HOST CLOCK TREE 0 on. Bit [1] Force HOST CLOCK TREE 0 off (Note 4-8). Bit [2] Force HOST CLOCK TREE 1 on. Bit [3] Force HOST CLOCK TREE 1 off. Bit [4] Force INTERRUPT AGGREGATOR CLOCK TREE on. Bit [5] Force INTERRUPT AGGREGATOR CLOCK TREE off. Bit [6] Force GPIO CLOCK TREE on. Bit [7] Force GPIO CLOCK TREE off. Note: Bits [7:4] have no affect when EC CLOCK TREE 0 is off.
Test Register	2Fh	MCHP Test Mode Register, Reserved for Microchip Bit [6:0] MCHP Reserved Bit [7] VREG Power-down Mode = 0 (default): 1.8V Regulator always powered. = 1: Suspend 1.8V Regulator in SYSTEM DEEPEST SLEEP state (see Table 5-8, “EC Controlled Dynamic Power States,” on page 84).

Note 4-7 The SerIRQ Mode bit controls the SER_IRQ pin, the CLKRUN# pin and the affects of LPC DMA requests on CLKRUN# (See Section 6.7, on page 120 and Section 6.8, on page 123.

Note 4-8 Forcing HOST CLOCK TREE 0 ‘off’ inhibits access to the Clock Tree Control 1 register. nSYS_RST to required to re-enable access to this register.

5.0 POWER, CLOCKS AND RESETS

5.1 General Description

The Power, Clocks and Resets chapter includes descriptions of the MEC1609/MEC1609i Clock Generator, Power Configuration and Reset Interface. The Clock Generator, in addition to describing clock sources, also features a Generic Block Clocking Model and a Power Management Interface. The Reset Interface description includes internal and external reset sources, as well as descriptions of an internal 1.8V Regulator and Power Mux. Other descriptions in this chapter include References, a Port List, Interrupt Interface and a Registers Interface.

The Power Configuration, Clock Generator and Reset circuits have the following features:

5.1.1 Power Configuration

- Description of Power Supplies and Clocks ACPI Context
- Enumerated Power Supply Configurations
- Power-Up Sequence Definition
- 1.8V Regulator
- Power Mux

5.1.2 Clock Generator

- Three Asynchronous Clock Sources: 64.52 MHz Ring Oscillator, 32.768 KHz Crystal Oscillator and PCI_CLK
- Efficient Logic Design and Controllable Master Clock Trees to Minimize Power Consumption
- Independent EC-driven Power Management Interface
- 64.52 MHz Ring Oscillator Optimized for 115.2K baud 16C550A UART Support
- Generic Block Clocking Model
- EC-accessible Registers Interface

5.1.3 Reset Interface

- VTR and VBAT Reset Signaling (VTRGD, VBAT_POR, nSYS_RST, nEC_RST)
- VCC Reset Signaling (VCC Power Good)
- Watch-Dog Timer Forced Reset
- Interrupt Interface
- Registers Interface

5.2 References

1. Advanced Configuration and Power Interface Specification, Revision 1.0b, February 2, 1999.
2. Intel® 82801DBM I/O Controller Hub 4 Mobile (ICH4-M), Datasheet, Order Number: 252337-001, Intel Corp., January 2003.
3. PCI Mobile Design Guide, Version 1.1, PCI-SIG, December 18, 1998.

5.3 Port List

The Port List shown in Table 5-1 is preliminary and subject to change.

TABLE 5-1: Power, Clocks and Resets Port List

Signal Name	Direction	Source	Destination	Description
ARC_CLK_DISABLE	Input	External - Embedded Controller Core (EC)	Internal - EC Power State Controls	Indication for the Power Management Interface (see Section 5.4.7 on page 83) that an EC Sleep instruction has occurred and the processor is sleeping or halted.
SLEEP_STATE	Output	Internal - EC Power State Controls	Internal - 64.52 MHz Ring Oscillator Control and External Functions as Needed.	System Sleeping State status indicator as described in "EC Power State Controls," on page 87.
SLEEP_FLAG	Output	Clock Control Register	Internal - 64.52 MHz Ring Oscillator Control, Block Sleep Enables and External Functions as Needed.	Sleep indicator from the Clock Control Register. See also "EC Power State Controls," on page 87 and "Block Sleep Enables," on page 89.
PCI_CLK	Input	External	Internal Test Functions	33 MHz PCI Clock Input (also TEST_CLK_IN).
LRESET#	Input	External	Internal	PCI Reset. See Section 5.6.9, "LPC RESET," on page 99.
XTAL1	Input	External	Internal - 32.768 KHz Crystal Oscillator	32.768 KHz Crystal Oscillator crystal input pin.
XTAL2	Output/Input	External	Internal - 32.768 KHz Crystal Oscillator	32.768 KHz Crystal Oscillator crystal output/single-ended clock source input pin (see XOSEL).
WAKE	Input	External	Internal - Wake Interface	Aggregated Wake indicator from the Section 16.0, "EC Interrupt Aggregator," on page 251 for the Power Management Interface (see Section 5.4.7 on page 83).
WDT_ALRT	Input	External	Internal - Reset Interface	Causes a Watch-Dog Timer Forced Reset as described in Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99.
XOSEL	Output	Internal	Internal - 32.768 KHz Crystal Oscillator and other functions as needed.	Clock Enable Register bit D0, used to select a singe-ended external input for the 32.768 KHz Crystal Oscillator ("XOSEL," on page 112).
ALL BLOCK SLEEP ENABLE OUTPUTS	Output	Internal - Block Sleep Enables	External - Sleep-able Blocks	To all blocks as defined in Section 5.7.5, "Block Sleep Enable Registers," on page 105. See also "Block Sleep Enables," on page 89
MCLK, EC_BUS_CLK_EN, LPC_BUS_CLK_EN, MCLK_DIV2_EN, MCLK_DIV4_EN, MCLK_DIV8_EN, MCLK_DIV16_EN, MCLK_DIV32_EN, MCLK_DIV64_EN, MCLK_DIV128_EN, MCLK_DIV640_EN, MCLK_DIV64_EN_HST, X32K_CLK	Output	Internal - Ring Oscillator Sourced Clocking and 32K Clock Domain	External	See Section 5.4.8, "Ring Oscillator Sourced Clocking," on page 91 and Section 5.4.9, "32K Clock Domain," on page 92.

TABLE 5-1: Power, Clocks and Resets Port List **(CONTINUED)**

Signal Name	Direction	Source	Destination	Description
32KHZ_OUT	Output	Internal	External	Off-Chip 32.768KHz Oscillator Output (see "32KHz OUTPUT," on page 101).
ALL BLOCK "CLOCK REQUIRED" STATUS BITS	Input	External - Sleep-able Blocks	Internal - 64.52 MHz Ring Oscillator Control	see Section 5.7.6, "Clock Required Status Registers," on page 108.
nSIO_RESET	Output	Internal	Internal/ External	EC-driven SIO Reset and External System Reset (nRESET_OUT-See Table 2-15, "Miscellaneous Functions," on page 17). (see "iRESET OUT," on page 104).
VCC_PWRGD	Input	External	Internal -Host Clock Domain	VCC Power Good Input. See Section 5.4.8.4, "Host Clock Domain," on page 92 and Section 5.6.8, "VCC Power Good," on page 99. The EC can determine the state of the VCC_PWRGD signal using VCC PWRGD bit in the PCR Status and Control Register. See also Section 5.6.9, "LPC RESET," on page 99.
VCC_PWRGD_BUFF	Output	Internal	External - Pad Buffers	Buffered VCC_PWRGD output used to tri-state VCC-related Pads.
PCR_INT	Output	Internal	External ("EC Interrupt Aggregator," on page 251)	see Section 5.9, "Interrupt Interface," on page 112.
FLASH_PGM	Input	External	Internal - Power-Fail and Reset Status Register	see "FLASH," on page 111.
LPC_RST#	Output	Internal	External (LPC Interface)	see Section 5.6.9, "LPC RESET," on page 99.
VTR	Power Well	External	–	Suspend Supply
VBAT	Power Well	External	–	Battery Supply
VSS	Power Well	External	–	Digital Ground
AGND	Power Well	External	–	Analog Ground for the 32.768 KHz Crystal Oscillator.
VTR_1.8	Power Well	Internal	–	Output of the internal 1.8 V regulator (see Section 5.6.6, "1.8V Regulator," on page 98).
VTR1.8_BAT	Power Well	Internal	–	Output of the internal Power Mux for VBAT-backed logic (see Section 5.6.7, "Power Mux," on page 98).
nSYS_RST	Output	Internal	Internal/ External	Synchronized VTR Power Good (Section 5.6.4, "nSYS_RST," on page 98).
nEC_RST	Output	Internal	Internal/ External	Stretched nSYS_RST used for EC reset and Registers Interface (see Section 5.6, "Reset Interface," on page 95).
VBAT_POR	Output	Internal	Internal/ External	VBAT Power On Reset (Section 5.6.3, "VBAT_POR," on page 98)
VR_CAP	Power Well	Internal	–	Capacitor Connection for Internal Voltage Regulator (4.7µF ±20%, ESR 2 Ohms, max.) (see also Section 5.6.6, "1.8V Regulator," on page 98).

5.4 Clock Generator

5.4.1 OVERVIEW

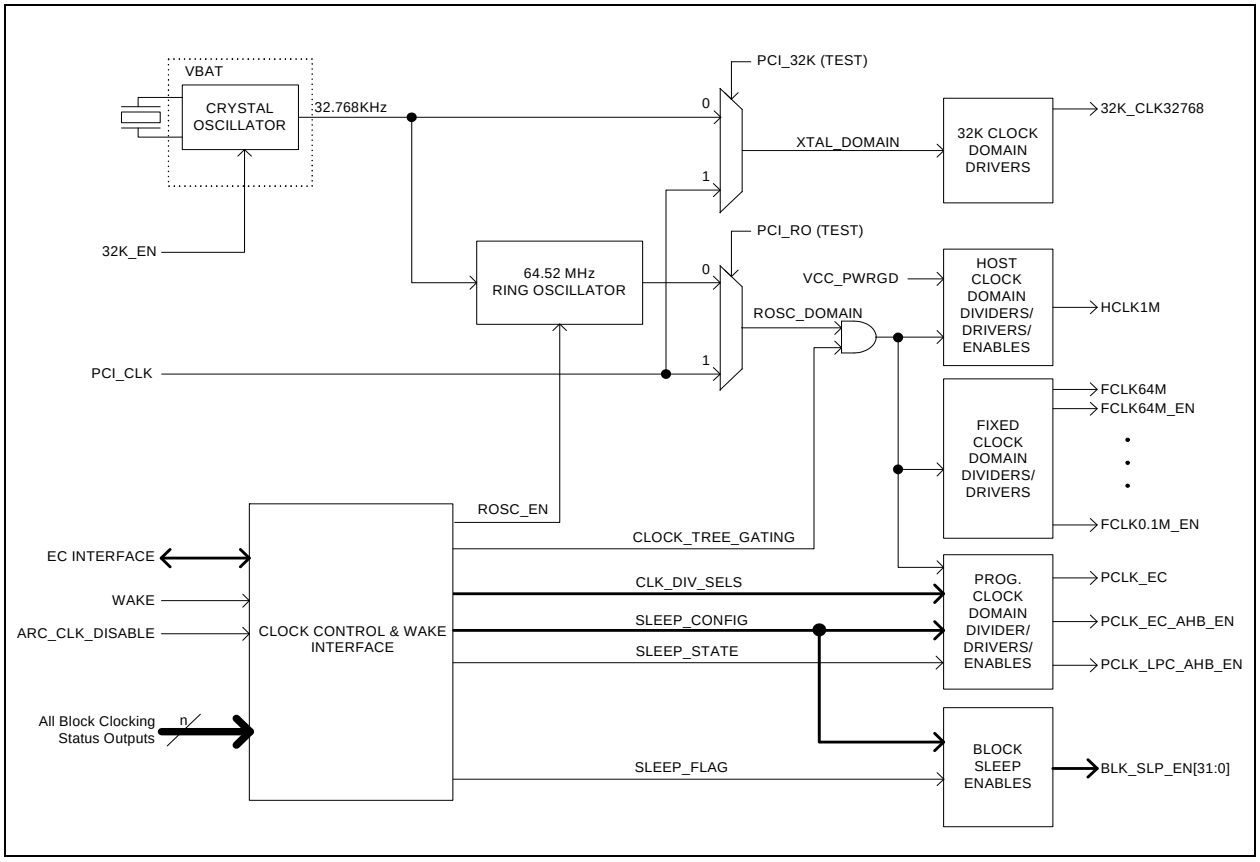
The MEC1609/MEC1609i Clock Generator includes three clock sources as illustrated in Figure 5-1, "Clock Generator Block Diagram": the 64.52 MHz Ring Oscillator, 32.768 KHz Crystal Oscillator and the PCI Clock (PCI_CLK in Table 5-1). The relationship of these clock sources to the system power supplies is described in Section 5.4.2, "Power Supplies and Clocking," on page 77; their relationship to the ACPI power states is described in Section 5.5.1, "Power Supplies and Clocks ACPI Context," on page 93.

Ring Oscillator Sourced Clocking includes Programmable Clock Domains, a Fixed Clock Domain and a Host Clock Domain. The output from the 32.768 KHz Crystal Oscillator defines a 32K Clock Domain.

The Clock Generator also includes the definition of a Generic Block Clocking Model that provides the foundation for a Power Management Interface. This interface defines several EC Controlled Dynamic Power States that can influence power consumption at the block level and within the Clock Generator.

The Clock Generator includes an EC accessible Registers Interface and support for an ATE test mode to drive all system clocking from the PCI_CLK input pin.

FIGURE 5-1: Clock Generator BLOCK DIAGRAM



5.4.2 POWER SUPPLIES AND CLOCKING

Table 5-2 illustrates clocking capabilities versus power supply availability. For more information, see Section 5.5, "Power Configuration," on page 93.

TABLE 5-2: CLOCKS VS. POWER SUPPLIES

Power Supply States (Note 5-1)			Clock			
VBAT	**VTR**	**VCC**	**PCI Clock**	**32K XTAL**	**32K External**	**Ring OSC.**
OFF	OFF	OFF	OFF	OFF	OFF	OFF
ON	OFF	OFF		ON/OFF (Note 5-2) OFF	ON/OFF (Note 5-4) OFF	
ON	ON	OFF				ON/OFF (Note 5-3, Note 5-5)
ON	ON	ON	ON/OFF			
OFF	ON	ON				

Note 5-1 power supply states not illustrated in Table 5-2 are undefined (see also Section 5.5, "Power Configuration," on page 93).

Note 5-2 this is true only after the EC asserts the 32K_EN bit in the Clock Enable Register.

Note 5-3 there is accuracy adjustment latency as described in Section 5.4.3, "64.52 MHz Ring Oscillator," on page 77. The 64.52 MHz Ring Oscillator can be disabled using the Power Management Interface.

Note 5-4 an external single-ended 32.768Hz clock source may be VTR or VBAT powered. Note that higher than normal VBAT current may occur when VTR transitions from unpowered to powered if the switching threshold on the external single-ended 32.768Hz clock source is different than the internal Power Mux switch threshold.

Note 5-5 the accuracy of the 64.52 MHz Ring Oscillator when the 32.768Hz clock source is 'off' is described in Section 5.4.3.

5.4.3 64.52 MHZ RING OSCILLATOR

The MEC1609/MEC1609i Clock Generator includes a high-accuracy, low power, low start-up latency 64.52 MHz Ring Oscillator. The 64.52 MHz Ring Oscillator is always enabled except in the SYSTEM DEEPEST SLEEP state when the 64.52 MHz Ring Oscillator is stopped by hardware as described in Section 5.4.7, "Power Management Interface," on page 83. The 64.52 MHz Ring Oscillator timing parameters are shown in Table 6.3.

Without correction or when the 32.768 KHz Crystal Oscillator is not running, the accuracy of the 64.52 MHz Ring Oscillator is between –50% and +2% of nominal value. When the 64.52 MHz Ring Oscillator is enabled and the 32.768 KHz Crystal Oscillator is running, the accuracy of the 64.52 MHz Ring Oscillator is automatically corrected by hardware to ±2% using a free-running iterative algorithm (see a description of the FREQ LOCK bit in the PCR Status and Control Register and the SAA bit in the Clock Control Register). The t_{ADJ} time is shown in Table 5-3.

The 64.52 MHz Ring Oscillator is reset by VTRGD as described in Section 5.6, "Reset Interface," on page 95.

TABLE 5-3: 64.52 MHz Ring Oscillator TIMING PARAMETERS

Parameters	Symbol	MIN	TYP	MAX	Units
Adjustment Delay to ±2% Accuracy	t_{ADJ}	0.03	-	4 (Note 5-6)	ms
64.52 MHz Ring Oscillator Start-up Delay (–50%/+2% accuracy)	t_{SU}			6	µs

Note 5-6 When the 32.768 KHz Crystal Oscillator is configured for use with a crystal (i.e., XOSEL is '0'), the 32.768 KHz Crystal Oscillator start-up time must be added to the t_{ADJ} time in Table 5-3 when the 32K_EN is asserted (see 32K_EN description in "32K_EN," on page 112).

5.4.4 32.768 KHZ CRYSTAL OSCILLATOR

The 32.768 KHz Crystal Oscillator provides a stable timebase for the 64.52 MHz Ring Oscillator and a clock source for the 32K Clock Domain. The XOSEL bit configures the 32.768 KHz Crystal Oscillator to use either a single-ended 32.768 KHz clock input or a 32.768 KHz crystal as described in "XOSEL," on page 112.

The 32.768 KHz Crystal Oscillator is controlled by the 32K_EN bit in the Clock Enable Register (see Section 5.8.2, "Clock Enable Register," on page 111). When XOSEL is not asserted and 32K_EN is asserted, there is a start-up delay (t_{SU}) for the 32.768 KHz Crystal Oscillator as shown in Table 5-4.

The clocks sourced by the 32.768 KHz Crystal Oscillator in the 32K Clock Domain operate as described in Table 5-17, "Typical MEC1609/MEC1609i Clocks vs. ACPI Power States" (see Section 5.5.1, "Power Supplies and Clocks ACPI Context," on page 93.

TABLE 5-4: 32.768 KHz Crystal Oscillator TIMING PARAMETERS

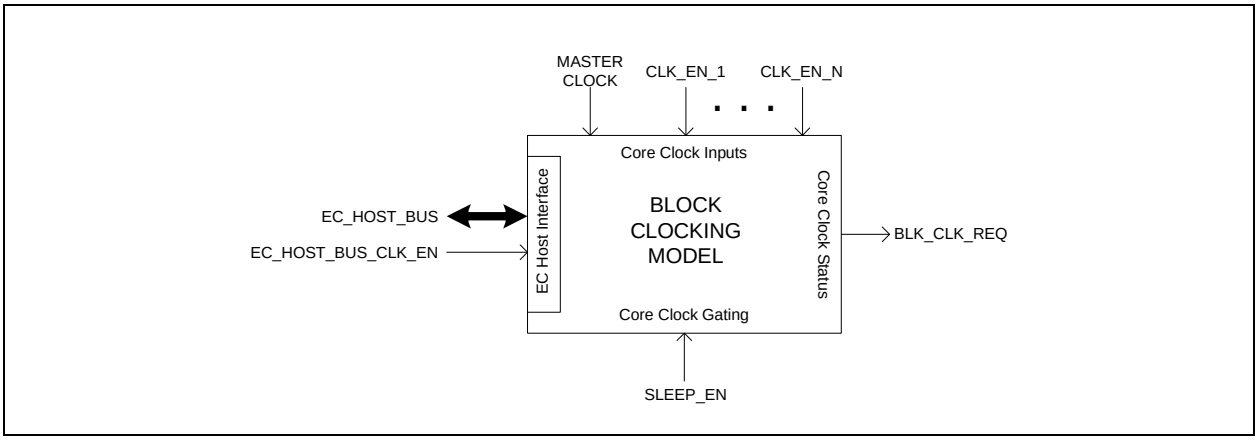
Parameters	Symbol	MIN	TYP	MAX	Units
32.768 KHz Crystal Oscillator Start-up Delay	t_{SU}			5	sec.
32.768 KHz Crystal Oscillator/Single-ended Clock Source Accuracy	Acc	See Application Note in Section 2.4.5, "Master Clock Interface," on page 14			

5.4.5 GENERIC BLOCK CLOCKING MODEL

5.4.5.1 Overview

The Generic Block Clocking Model defines the block Clock Gating interface that is assumed by the Clock Generator for all of the MEC1609/MEC1609i internal blocks identified in the Block Sleep Enable Registers. Components of this model are illustrated in Figure 5-2 and described in Section 5.4.5.2. The response of this model to the actions of the Power Management Interface is described in Section 5.4.5.3, "Behavior," on page 80.

FIGURE 5-2: Generic Block Clocking Model ILLUSTRATION



5.4.5.2 Components

As shown in Figure 5-2 the external interface for the Generic Block Clocking Model includes an EC (or other available bus master) host interface, core clock Inputs (which may be clocks or clock enables), a logical core clock gating control and a core clock status output. Not shown in Figure 5-2 is the internal interface for the Generic Block Clocking Model that includes a block enable bit and may also include a block idle status indicator. Each of the Generic Block Clocking Model internal and external interface elements and operational states are described in Table 5-5, "Generic Block Clocking Model Components".

When firmware de-asserts the internal block enable bit the block is disabled and in a minimum power consumption state. Depending on the implementation, the host may need to provide that the block is not in use before the internal enable bit is de-asserted because it may also function as a reset. Transitions to a minimum block power consumption

state while the internal enable bit remains asserted may be requested by the Power Management Interface using an external sleep enable input (see also Section 5.4.5.3, "Behavior," on page 80). In both cases (i.e., when the block is disabled or sleeping), the core clock required status indicator output (BLK_CLK_REQ in Figure 5-2) is de-asserted.

When firmware asserts the internal block enable and the external sleep enable input is not asserted, or the external sleep enable input is asserted but the internal idle indicator is not asserted, the block is operational and in a maximum power consumption state. In both of these cases, the core clock required status indicator output is asserted.

TABLE 5-5: Generic Block Clocking Model COMPONENTS

Internal Enable Bit (Note 5-8)	External SLEEP_EN Input (Note 5-9)	Block Idle Status	Core Clock Required Status Output (Note 5-7)	State	Power	Description
0	X	X	0	DISABLED	MINIMUM	Block is disabled by firmware and the core clock is not needed and gated 'off' internally. Note: it may be up to the host to provide that the block is not in use before the internal enable bit is de-asserted because the internal enable may also function as a reset when not asserted.
1	0	NOT IDLE	1	FULL POWER	MAXIMUM	The full power state identifies the block normal operation mode where the block is neither disabled by firmware nor commanded to sleep by the Power Management Interface.
		IDLE				
	1	NOT IDLE		PREPARING TO SLEEP		A sleep command has been asserted but the core clock is still required because the block is not idle.
		IDLE	0	SLEEPING	MINIMUM	A sleep command has been asserted, the block is idle and the core clocks are stopped (Note 5-10).

Note 5-7 the "Block Clock Required Status Output" (BLK_CLK_REQ in Figure 5-2) only reflects the core clock requirement; i.e. independent of the host interface clock enable (EC_HOST_BUS_CLK_EN in Figure 5-2). The MEC1609/MEC1609i Generic Block Clocking Model assumes that the block may not remain operational without the host interface clock enable which will not be stopped unless the 64.52 MHz Ring Oscillator is disabled (see Section 5.4.7, "Power Management Interface," on page 83). The "Clock Required Status" for each block can be seen in the Clock Required Status Registers.

Note 5-8 the internal enable bit (not shown in Figure 5-2) is accessible through the EC Host Interface shown in Figure 5-2 and provides a reset to each block. Typically, as soon as the internal enable bit is de-asserted, the block may be immediately reset and held in the lowest power consumption state.

Note 5-9 The external sleep enables are configured using the Block Sleep Enables as described in "Block Sleep Enables," on page 89.

Note 5-10 State transitions on the internal enable bit are undefined in the SLEEPING state and may produce undesireable results.

5.4.5.3 Behavior

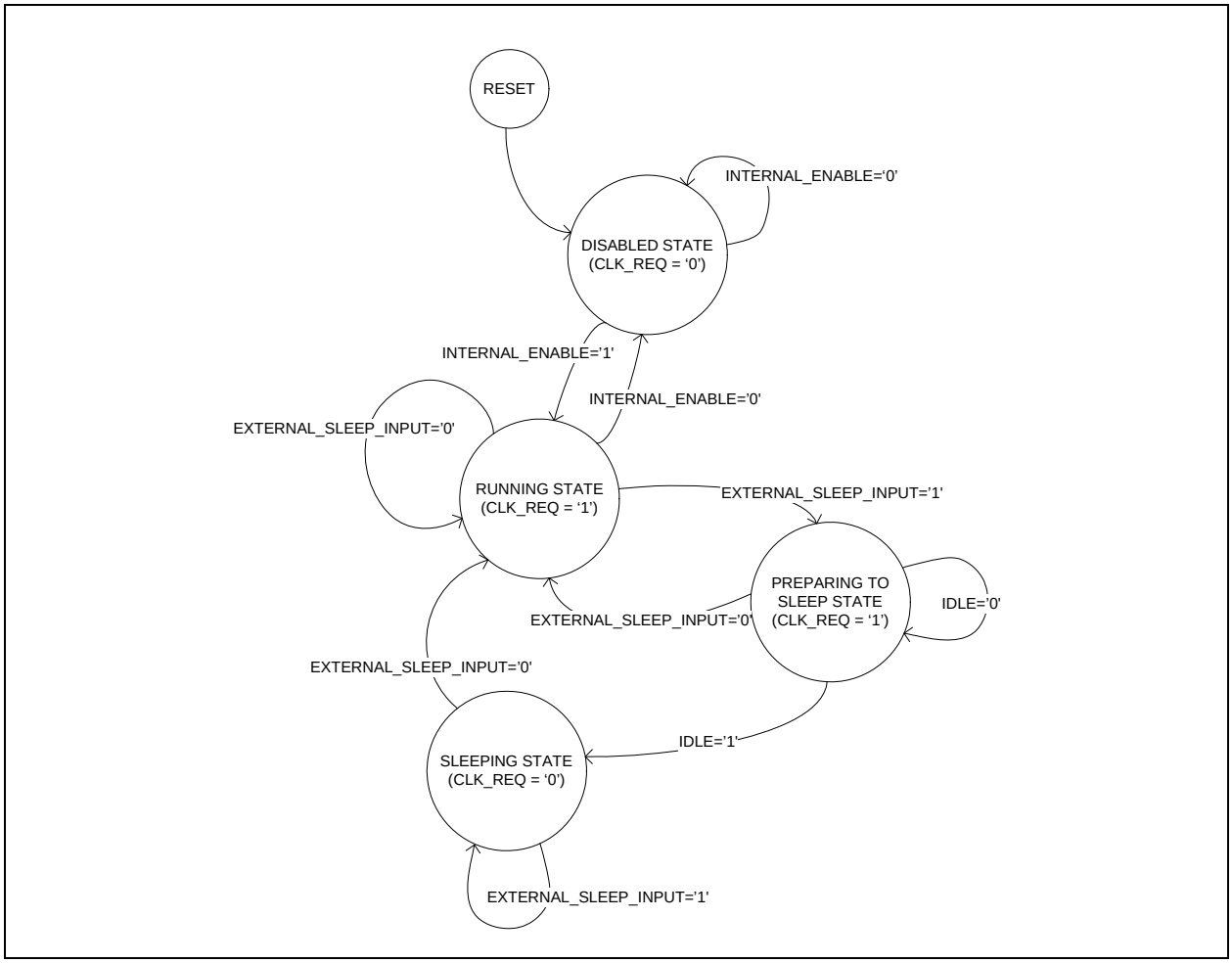
The affects of the Power Management Interface on the block sleep Behavior as a result of transitions on the sleep enable input are ignored when the block is disabled. Blocks are typically disabled following a power-on reset (Figure 5-4, "Mode Transitions for Dynamic Power Management").

When a block is enabled, a sleep state request as a result of a transition on the sleep enable input must not adversely affect block operation; e.g., by performing illegal operations on an external circuit or corrupting transaction data. As a result, there may be transition latency from the running state to the sleeping state, depending on the nature of the block and its operational state as defined by an internal block 'idle' indicator when the sleep enable is asserted. It is possible that a block may never enter the sleeping state if the block does not idle before the sleep enable input is de-asserted as a result of a wake event.

Once a block enters the sleeping state, internal clocks are gated 'off' and the block is inactive; i.e., outputs are static and the block cannot respond to transitions on inputs, except as defined by the Wake Interface. The transition from the sleeping state to the running state can only occur once the system clocks are running and the sleep enable input is de-asserted.

As described in Section 5.4.5.2, "Components," on page 78, transitions from the running state or the preparing to sleep state to the disabled state may occur without latency depending on the implementation (not shown in Figure 5-4). Transitions from the sleeping state to the disabled state are undefined when the 64.52 MHz Ring Oscillator is stopped. Transitions from the sleeping state to the disabled state can occur without latency when the 64.52 MHz Ring Oscillator is running (also not shown in Figure 5-4).

FIGURE 5-3: Generic Block Clocking Model CLOCK GATING STATE DIAGRAM EXAMPLE



5.4.6 MASTER CLOCK TREES

5.4.6.1 Description

The master clock derived from the 64.52 MHz Ring Oscillator is branched into six internal clock trees in the MEC1609/MEC1609i. These Master Clock Trees are controlled by register bits in the Clock Tree Control 0 and Clock Tree Control 1 registers and function as defined in Section 5.4.6.4, "Control Bit Encoding," on page 82 and Section 5.4.6.2, "Dynamic Clock Tree Gating," on page 81. The Block Allocation Per Clock Tree is defined in Section 5.4.6.3.

Each master clock tree can be forced 'on,' forced 'off,' or dynamically controlled by hardware to help minimize power consumption throughout the EC Controlled Dynamic Power States. See also Section 5.4.7.3, "Clock-tree Gating in Heavy Sleep," on page 85.

5.4.6.2 Dynamic Clock Tree Gating

5.4.6.2.1 Overview

Dynamic Clock Tree Gating specifies that a master clock tree is 'on' if any one of the blocks allocated to that tree requires a clock, or if a bus master is accessing registers in that tree; clock trees are 'off' when the bits in the Clock Required Status Registers that are associated with the blocks in a tree are not asserted and a bus master is not accessing registers in the tree.

Dynamic Clock Tree Gating is enabled as described in Table 5-7, "Clock Tree Force Control Bits Encoding," on page 82 and is enabled by default for the Master Clock Trees (see also Note 5-12).

The EC Interrupt Aggregator and the GPIO Interface are special cases as described in Section 5.4.6.2.2, "INTERRUPT AGGREGATOR CLOCK TREE" and Section 5.4.6.2.3, "GPIO CLOCK TREE".

5.4.6.2.2 INTERRUPT AGGREGATOR CLOCK TREE

The EC Interrupt Aggregator does not conform to the Generic Block Clocking Model, but instead dynamically gates its master clock 'off' when all source interrupts and wake-up events are *not* asserted. For the INTERRUPT AGGREGATOR CLOCK TREE to remain 'off,' all active interrupts must be cleared at the source.

For example, the PCR_INT interrupt is asserted following a VBAT_POR and must be cleared to enable Dynamic Clock Tree Gating in the INTERRUPT AGGREGATOR CLOCK TREE.

Additionally, when an interrupt from the GPIO Interface is configured as level-sensitive, this interrupt will be continuously asserted when the voltage at the pin changes to the specified level, which will also inhibit Dynamic Clock Tree Gating in this interface. In this instance, Dynamic Clock Tree Gating can only be achieved by setting the GPIO Interface interrupt(s) to edge detect mode.

As shown in Table 5-6, the EC Interrupt Aggregator is part of EC CLOCK TREE 0. Dynamic Clock Tree Gating in EC CLOCK TREE 0 cannot occur until the INTERRUPT AGGREGATOR CLOCK TREE is gated 'off.'

5.4.6.2.3 GPIO CLOCK TREE

The GPIO Interface does not conform to the Generic Block Clocking Model, but instead dynamically gates its master clock 'off' when all GPIO Interface interrupts and wake-up events are *not* asserted. To enable Dynamic Clock Tree Gating for the GPIO CLOCK TREE, follow the procedure described for the INTERRUPT AGGREGATOR CLOCK TREE to set the GPIO interrupts to edge detect mode.

As shown in Table 5-6, the GPIO Interface is part of EC CLOCK TREE 0. Dynamic Clock Tree Gating in EC CLOCK TREE 0 cannot occur until the GPIO CLOCK TREE is gated 'off.'

5.4.6.3 Block Allocation Per Clock Tree

The Block Allocation Per Clock Tree is defined in Table 5-6.

Note that clocking for the ARC 625D Embedded Controller is also dynamically controlled as defined in Section 5.4.7, "Power Management Interface," on page 83, but is not considered one of the Master Clock Trees as described here; however, EC CLOCK TREE 0 remains 'on' whenever the ARC 625D Embedded Controller clock is running.

TABLE 5-6: MASTER CLOCK TREE BLOCK ALLOCATION

	Clock Tree					
	EC CLOCK TREE 0	**EC CLOCK TREE 1**	**EC CLOCK TREE 2**	**EC CLOCK TREE 3**	**HOST CLOCK TREE 0**	**HOST CLOCK TREE 1**
BLOCKS	Internal Busses	16-Bit Timer Interface	PS/2 Device Interface	Serial Debug Port	Logical Device Configuration Global Register Bank	MailBox Register Interface
	EC Interrupt Aggregator	BC-Link Master	PWM Controller	Watchdog Timer Interface Register Bank	Embedded Flash Subsystem Host Controller	ACPI PM1 Block Interface
	LED Interface	VBAT Register Bank	TACH Monitor	EC General Purpose Serial Peripheral Interface (GP-SPI)	Host Interface (LPC)	VLPC Bus Interface
	GPIO Interface	Hibernation Timer Register Bank	Keyboard Matrix Scan Support	RC Identification Detection (RC_ID)	MCHP Reserved	Two Pin Serial Port (UART)
	ARC SRAM	SMB Device Interface	Analog to Digital Converter (ADC)	PECI Interface	ACPI Embedded Controller Interface	Host General Purpose Serial Peripheral Interface (GP-SPI)
	Power, Clocks and Resets Register Bank	–	–	Input Capture and Compare Timer	8042 Emulated Keyboard Controller	–
	–	–	–	DMA Controller	Embedded Memory Interface	–
	–	–	–	EC AHB SPI Flash Read Controller	–	–
	–	–	–	JTAG and Boundary Scan	–	–

Note 5-11 EC CLOCK TREE 0 remains ‘on’ whenever the ARC 625D Embedded Controller clock is running. None of the blocks allocated to EC CLOCK TREE 0 have “Clock Required” Power Management Interface outputs. EC CLOCK TREE 0 remains ‘off’ whenever the ARC 625D Embedded Controller is sleeping.

5.4.6.4 Control Bit Encoding

TABLE 5-7: CLOCK TREE FORCE CONTROL BITS ENCODING

Force ‘OFF’ Bit	Force ‘ON’ Bit	Description
0	0	Dynamic Clock Tree Gating enabled (Default) (Note 5-12).
0	1	Clock Tree Forced ‘On’
1	0	Clock Tree Forced ‘Off’
1	1	

Note 5-12 to shut down EC CLOCK TREE 0, HOST CLOCK TREE 0 and HOST CLOCK TREE 1 drive the VCC_PWRGD input pin to ground while the VCC_PWRGD signal function is selected (the default setting for this pin).

5.4.7 POWER MANAGEMENT INTERFACE

5.4.7.1 Overview

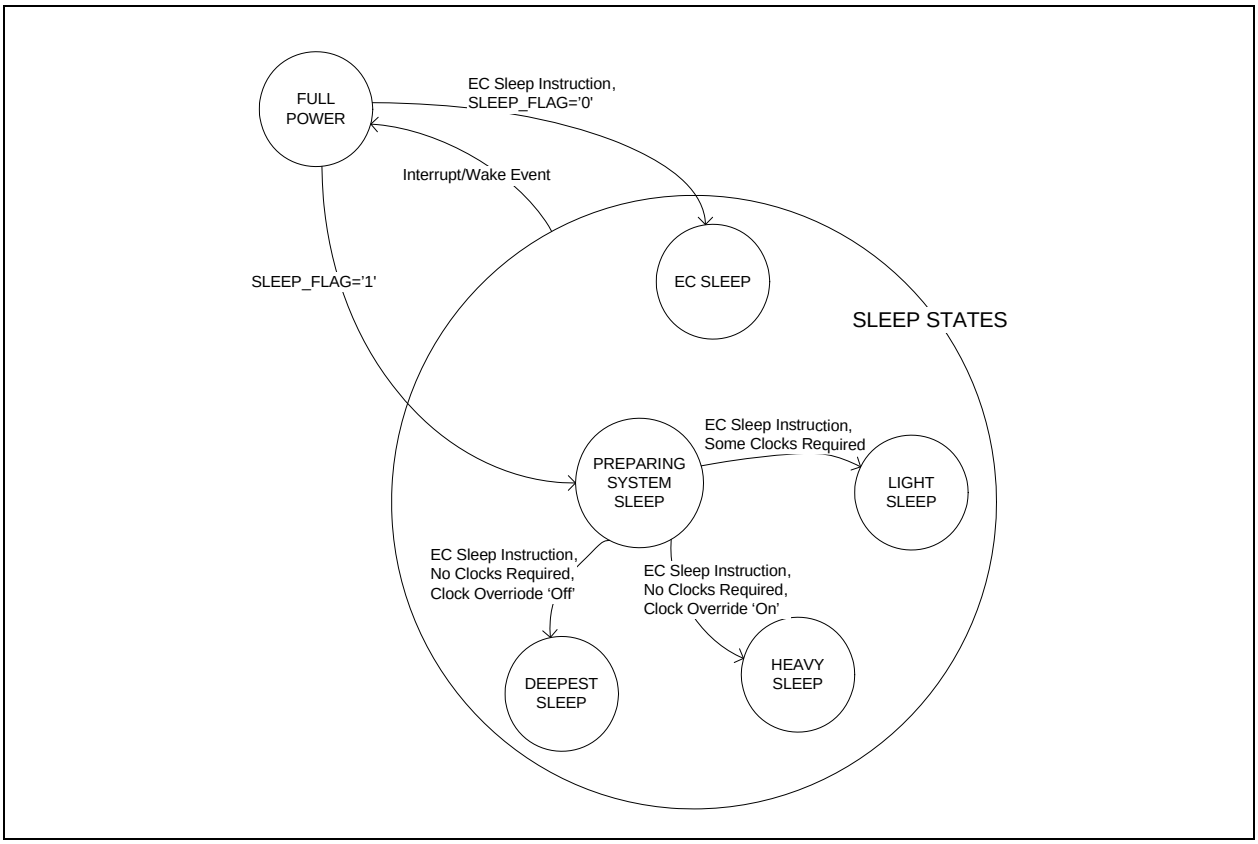
The MEC1609/MEC1609i includes several features to help minimize power consumption, the most intrinsic of which is the application of advanced gate-level low-power design techniques. The EC can also establish the upper run-time power consumption limit by asserting individual internal block enables (i.e., as described in the Generic Block Clocking Model) only for functions that are absolutely required during normal operation. Finally, the EC can also dynamically minimize power consumption by modulating clocks at the block level and within the Clock Generator using the clock gating feature of the Power Management Interface.

APPLICATION NOTE: implementing dynamic power management using individual internal block enables alone is complicated by the fact that, depending on the block implementation, block enables may also perform a reset function.

There are six EC Controlled Dynamic Power States as described in Section 5.4.7.2, "EC Controlled Dynamic Power States". These states are achieved using clock gating as described in Section 5.4.7.5, "Clock Gating," on page 87, which can also affect the 64.52 MHz Ring Oscillator.

Running and sleeping are the two basic operational modes when considering dynamic power management as defined by this interface (Figure 5-4). Transitions between these modes are deliberate and persistent. For example, to exit the full power state the EC must issue a sleep command as described in "EC Controlled Sleep State Activation," on page 88. Exiting sleep states requires the Wake Interface.

FIGURE 5-4: MODE TRANSITIONS FOR DYNAMIC POWER MANAGEMENT



5.4.7.2 EC Controlled Dynamic Power States

Table 5-8 illustrates the EC Controlled Dynamic Power States that can be achieved using the clock gating feature of Power Management Interface. The EC Controlled Dynamic Power States closely mirror the system power states defined by the Generic Block Clocking Model, but 1) redefine the “preparing to sleep” state to include the affects of the EC sleep state, 2) define additional implementation-specific sleep states that affect Wake Interface latency and 3) indicate the aggregated response of all the MEC1609/MEC1609i power-managed blocks. Typically, the higher the state number in Table 5-8. the greater the system power savings. Traversing the EC Controlled Dynamic Power States requires the SLEEP_FLAG**,** ARC_CLK_DISABLE signal, WAKE signal, the ROSC_SLP_OVRD bit, the Block Sleep Enable Registers and the Clock Required Status Registers

See Section 40.4, "Power Consumption," on page 504 or characterization current values.

TABLE 5-8: EC CONTROLLED DYNAMIC POWER STATES

Device Power State	SLEEP_FLAG (Note 5-13)	ARC_CLK_DISABLE (Note 5-14)	Global Block Clock Status (Note 5-15)	Ring OSC. Sleep OVRD. (Note 5-16)	State Name	Description
0.	0	0	X	X	FULL POWER	The system is running. This is the highest power consumption state.
1.	0	1	X	X	EC SLEEP	EC has executed a sleep instruction. The rest of the system is unaffected by the EC SLEEP state.
2.	1	0	X	X	PREPARING SYSTEM SLEEP	Sleep commands issued to all sleep-enabled blocks (see "EC Controlled Sleep State Activation," on page 88). The EC can return to the FULL POWER state from PREPARING SYSTEM SLEEP by de-asserting (‘0’) SLEEP_FLAG (see also Note 5-17).
3.	1	1	Clock Required	X	SYSTEM LIGHT SLEEP	System is in a sleeping mode but the 64.52 MHz Ring Oscillator must remain operating because one or more blocks require a clock.
4.	1	1	Clock Not Required	1	SYSTEM HEAVY SLEEP	System is in a sleeping mode and no blocks require the clock but the 64.52 MHz Ring Oscillator must remain operating because the EC decided to limit start-up latency by asserting the ROSC_SLP_OVRD which prevents the 64.52 MHz Ring Oscillator from turning off.
5.	1	1	Clock Not Required	0	SYSTEM DEEPEST SLEEP	System is in a sleeping mode, no blocks require the clock and the 64.52 MHz Ring Oscillator is stopped. This is the lowest possible power consumption state that can be achieved using the Power Management Interface.

Note 5-13 SLEEP_FLAG is bit D1 in the Clock Control Register.

Note 5-14 the ARC_CLK_DISABLE signal is described in Table 5-1, “Power, Clocks and Resets Port List,” on page 74.

Note 5-15 includes the sum of all the “Core Clock Required Status” outputs defined in the Generic Block Clocking Model and aggregated in the Clock Required Status Registers.

Note 5-16 this column refers to the ROSC_SLP_OVRD bit in the Clock Control Register.

Note 5-17 In the PREPARING SYSTEM SLEEP state when the sleep enable to a block is asserted, EC register accesses to that block are undefined and should be avoided.

5.4.7.3 Clock-tree Gating in Heavy Sleep

The output of the 64.52 MHz Ring Oscillator is gated 'off' in the System Heavy Sleep State as defined in Table 5-9 to conserve power beyond gating the Master Clock Trees as described in Section 5.4.6.

TABLE 5-9: MEC1609/MEC1609I Clock-tree Gating in Heavy Sleep IN HEAVY SLEEP STATE

Power States	Clock Required Status Registers	ROSC_SLP_ OVRD Bit	Clock-tree Gating in Heavy Sleep	Description
FULL POWER	N/A	N/A	NO	The system is running.
EC SLEEP	N/A	N/A	NO	Only the EC is sleeping.
PREPARING SYSTEM SLEEP	N/A	N/A	NO	Sleep commands issued to all blocks, but the EC has not executed a Sleep Instruction.
SYSTEM LIGHT SLEEP	Some blocks require a clock.	N/A	NO	Sleep commands issued to all blocks and the EC is sleeping.
SYSTEM HEAVY SLEEP	No blocks require a clock.	Asserted (Override ROSC sleep)	YES	Sleep commands issued to all blocks, the EC is sleeping and no blocks require a clock, but the 64.52 MHz Ring Oscillator remains running to limit start-up latency. Clock-tree Gating in Heavy Sleep is enabled to conserve power.
SYSTEM DEEPEST SLEEP	No blocks require a clock.	Not Asserted (Do Not Override ROSC sleep)	NO	Clock-tree Gating in Heavy Sleep is not required because the 64.52 MHz Ring Oscillator is 'off.'

5.4.7.4 64.52 MHz Ring Oscillator Control

The 64.52 MHz Ring Oscillator can only be controlled by the EC using the Power Management Interface. As illustrated in Figure 5-5, "64.52 MHz Ring Oscillator Controls" and described in Table 5-10, "ROSC_EN Control", the 64.52 MHz Ring Oscillator remains running whenever a single block "Core Clock Required Status" output is asserted, the ROSC_SLP_OVRD is asserted, SLEEP_STATE is not asserted, or when WAKE is asserted. The 64.52 MHz Ring Oscillator will only stop when all block "Core Clock Required Status" outputs and the ROSC_SLP_OVRD are not asserted, WAKE is not asserted and SLEEP_STATE is asserted.

FIGURE 5-5: 64.52 MHZ RING OSCILLATOR CONTROLS

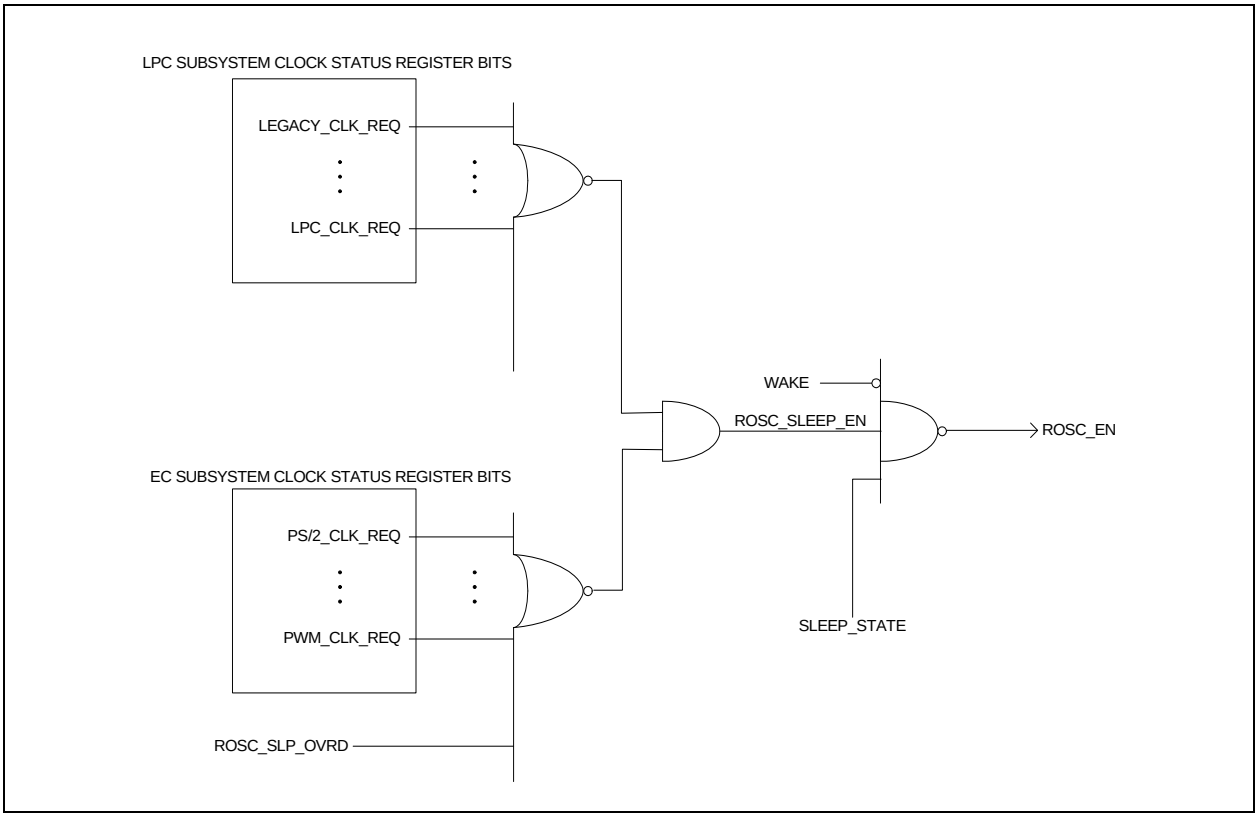


TABLE 5-10: ROSC_EN CONTROL

ROSC_SLEEP_ Enable (Note 5-18)	SLEEP_STATE (Note 5-19)	Wake (Note 5-20)	ROSC_EN (Note 5-21)	Description
0	X	X	1	The 64.52 MHz Ring Oscillator is always enabled (“on”) when at least one block requires the clock or the EC has overridden the 64.52 MHz Ring Oscillator sleep state using the ROSC_SLP_OVRD bit in the Clock Control Register.
X	0	X		The 64.52 MHz Ring Oscillator is always enabled (“on”) when SLEEP_STATE is not asserted.
X	X	1		The 64.52 MHz Ring Oscillator is always enabled (“on”) when WAKE is asserted.
1	1	0	0	64.52 MHz Ring Oscillator is disabled (“off”) when SLEEP_STATE is asserted, all blocks do not require the 64.52 MHz Ring Oscillator, the EC has not overridden the 64.52 MHz Ring Oscillator sleep state using the ROSC_SLP_OVRD bit and WAKE is not asserted.

Note 5-18 ROSC_SLEEP_ENABLE is illustrated in Figure 5-5, "64.52 MHz Ring Oscillator Controls" and indicates the status of all the “Core Clock Required Status” outputs defined in the Generic Block Clocking Model and aggregated in the Clock Required Status Registers and the ROSC_SLP_OVRD bit. For a description of the ROSC_SLP_OVRD signal, see "ROSC_SLP_OVRD," on page 101.

Note 5-19 see "EC Power State Controls," on page 87 and Table 5-1 for a description of the SLEEP_STATE signal.

Note 5-20 see Section 5.4.7.6, "Wake Interface," on page 90 and Table 5-1 for a description of the WAKE signal.

Note 5-21 ROSC_EN is the 64.52 MHz Ring Oscillator enable control illustrated in Table 5-5.

5.4.7.5 Clock Gating

5.4.7.5.1 Overview

Power savings using the Power Management Interface comes from Clock Gating as described above and in the sub-sections that follow. The magnitude of the power savings depends on the configuration of the Block Sleep Enables, the ROSC_SLP_OVRD bit in the Clock Control Register, the configuration of the Master Clock Trees and the operational status of the individual blocks as defined by the Clock Required Status Registers.

5.4.7.5.2 EC Power State Controls

Assuming WAKE is not asserted, when the SLEEP_FLAG bit in the Clock Control Register is asserted by the EC (see also "EC Controlled Sleep State Activation," on page 88), a sleep command is sent to all blocks that are configured for sleep as defined by the Block Sleep Enables. Clock Gating occurs at the block level following the sleep command depending on the state of the block clocking requirement (see Section 5.7.6, "Clock Required Status Registers," on page 108).

Once the SLEEP_FLAG bit is asserted, Clock Gating can only occur within the Clock Generator when the ARC_CLK_DISABLE signal (see Table 5-1, “Power, Clocks and Resets Port List,” on page 74) is asserted, at which time the SLEEP_STATE signal that can affect 64.52 MHz Ring Oscillator Control is asserted (Table 5-11). Timing for the EC Power State Controls is defined in Figure 6.7 and Table 6.11.

If WAKE is asserted when EC Controlled Sleep State Activation is attempted, the 64.52 MHz Ring Oscillator will remain enabled, the SLEEP_FLAG will be automatically de-asserted by hardware as described in Section 5.4.7.6, "Wake Interface," on page 90 and an EC interrupt will occur (not shown in Table 5-11).

TABLE 5-11: EC Power State Controls DESCRIPTION

ARC_CLK_DISABLE (Note 5-14)	SLEEP_FLAG (Note 5-13)	SLEEP_STATE (Note 5-23)	States (Note 5-22)	Description
0	0	0	FULL POWER	–
0	1	0	PREPARING SYSTEM SLEEP	SLEEP_FLAG has been asserted by the EC which asserts sleep enables to blocks that have been enabled for sleeping; i.e., clocks in these blocks are turning ‘off’ in this state. The EC and the 64.52 MHz Ring Oscillator clock remain active. The PREPARING SYSTEM SLEEP state may represent a lower than FULL POWER system power consumption state.
1	0	0	EC SLEEP	The Clock Generator is unaffected in the EC SLEEP state. The EC SLEEP state may or may not represent a lower system power state than the PREPARING SYSTEM SLEEP state depending on the Block Sleep Enables and the system operational state.
1	1	1	SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP, SYSTEM DEEPEST SLEEP	The 64.52 MHz Ring Oscillator may be stopped as described in Section 5.4.7.4, "64.52 MHz Ring Oscillator Control," on page 85 and in Table 6.7.

Note 5-22 see the system state definitions in Section 5.4.7.2, "EC Controlled Dynamic Power States," on page 84.

Note 5-23 SLEEP_STATE affects the 64.52 MHz Ring Oscillator as described in Section 5.4.7.4, "64.52 MHz Ring Oscillator Control," on page 85 and appears in Table 5-1, “Power, Clocks and Resets Port List,” on page 74.

FIGURE 5-6: EC Power State Controls TIMING

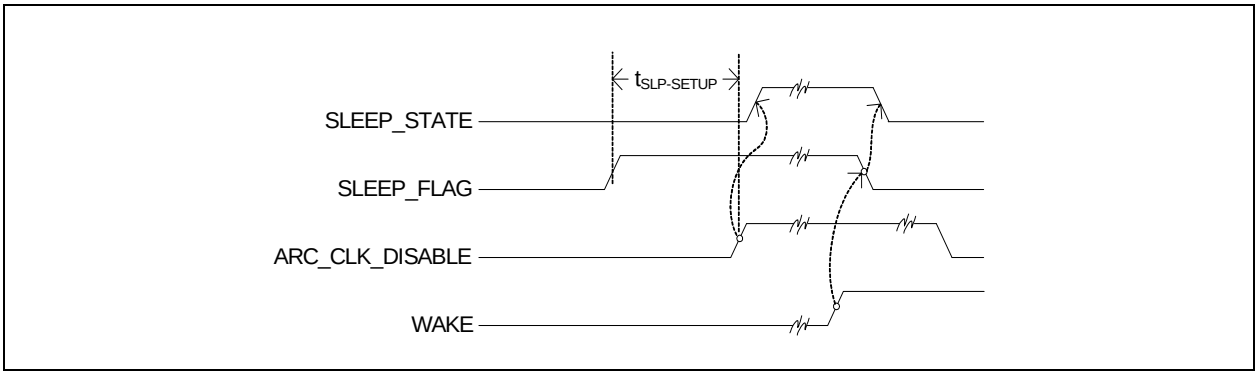


TABLE 5-12: EC Power State Controls TIMING PARAMETERS

Parameters	Symbol	MIN	TYP	MAX	Units
System Sleep Setup Time	$t_{SLP-SETUP}$	1	–	–	EC Clock

5.4.7.5.3 EC Controlled Sleep State Activation

OVERVIEW

EC Controlled Sleep State Activation depends upon the assertion of the ARC_CLK_DISABLE signal which occurs when the EC executes a sleep instruction. The dynamic sleep states that can be achieved through EC Controlled Sleep State Activation in part depend on the state of the SLEEP_FLAG in the Clock Control Register as defined in the sections below, “Entering and Exiting EC Sleep State” and “Entering and Exiting System Sleep States.” Simultaneous assertions of the WAKE input and EC Controlled Sleep State Activation immediately terminate system sleeps states as defined in Section 5.4.7.6, "Wake Interface," on page 90.

ENTERING AND EXITING EC SLEEP STATE

As illustrated in Section TABLE 5-8:, "EC Controlled Dynamic Power States," on page 84, transitions to the EC SLEEP state occur when the ARC_CLK_DISABLE signal is asserted (i.e., the EC enters the sleep state) while the SLEEP_FLAG in the Clock Control Register is not asserted (‘0’). The EC SLEEP state is terminated when an interrupt to the EC occurs, as described in Section 5.4.7.6, "Wake Interface," on page 90.

The EC enters the sleep state when the SLEEP instruction is issued. The SLEEP instruction halts the CPU pipeline and gates the processor clocks. The clocks to the ARC interrupt logic are not gated. As long as interrupts are enabled in the ARC core STATUS register and at least one interrupt is enabled in the Interrupt Accelerator when the SLEEP instruction is issued, the processor will wake up and process the interrupt service routine when the interrupt triggers. On return from interrupt, the processor will execute the instruction immediately following the SLEEP instruction.

If it is necessary to enable a wakeup interrupt just before entering the sleep state, some care is required to insure that the interrupt does not fire before the SLEEP instruction is issued. The ARC FLAG instruction should be used to enable and disable interrupts and the FLAG and SLEEP instructions should be contiguous to insure that the SLEEP is in the processor pipeline when interrupts are enabled and therefore executes before an interrupt can fire. If the code is executed from the instruction cache, the two instructions must be in the same cache line. The following example illustrates the code sequence:

```
.equ EI,0x06            ; Constant to enable both interrupt levels
.align 8                ; ensure cache alignment is to 8 bytes
FLAG EI                 ; Enable interrupts
SLEEP                   ; Put processor into sleep mode
```

For more information on the FLAG and SLEEP instructions, see the *ARCompact™ ISA Instruction Set Architecture Programmer's Reference*.

ENTERING AND EXITING SYSTEM SLEEP STATES

As illustrated in Section TABLE 5-8:, "EC Controlled Dynamic Power States," on page 84, transitions to the system sleep states (i.e., sleep states other than EC SLEEP) occur when the ARC_CLK_DISABLE signal is asserted (i.e., the EC enters the sleep state) while the SLEEP_FLAG in the Clock Control Register is asserted ('1'). These states include SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP and SYSTEM DEEPEST SLEEP.

The system sleep states are terminated when a wake event occurs as described in Section 5.4.7.6, "Wake Interface," on page 90 Entering and Exiting System Sleep States is similar to Entering and Exiting EC Sleep State. Refer to the section Entering and Exiting EC Sleep State for information about sleeping the EC.

Note that as described in Section 5.4.7.2, "EC Controlled Dynamic Power States," on page 84, the PREPARING SYSTEM SLEEP state in Table 5-8 occurs as soon as the SLEEP_FLAG is asserted, independent of the state of ARC_CLK_DISABLE. The EC can optionally interrogate the Clock Required Status Registers to estimate the depth of the sleep state, for example, before executing a sleep instruction. The EC can return to the FULL POWER state from the PREPARING SYSTEM SLEEP state at any time before executing a sleep instruction by de-asserting the SLEEP_FLAG.

When the ring oscillator is shut down, the oscillator can only be restarted by the Wake Interface or by power cycling the system. In order for the Wake Interface to operate, at least one wake capable interrupt must be enabled in the Interrupt Aggregator. This class of interrupts is described in Section 16.3.5, "Wake Capable Interrupts," on page 258.

5.4.7.5.4 Block Sleep Enables

The Block Sleep Enables allow the EC firmware to determine which blocks will receive sleep commands as a result of EC Controlled Sleep State Activation (see "Entering and Exiting System Sleep States," on page 89). The Block Sleep Enables are configured using the Block Sleep Enable Registers (see Section 5.7.5, "Block Sleep Enable Registers," on page 105) and behave as illustrated in Figure 5-7, "Block Sleep Enables Example" and Table 5-13, "Block Sleep Enables Definition". There are three Block Sleep Enable Registers: two EC Blocks Sleep Enables Registers (see Section 5.7.5.2 on page 106) and one LPC Blocks Sleep Enables Register (Section 5.7.5.1 on page 105).

FIGURE 5-7: Block Sleep Enables EXAMPLE

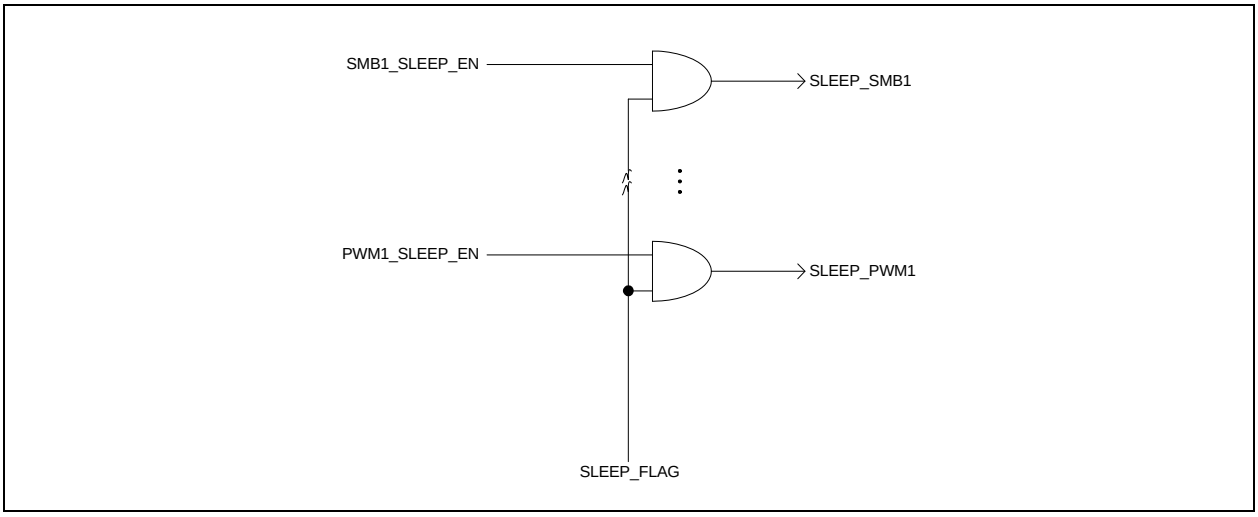


TABLE 5-13: Block Sleep Enables DEFINITION

Sleep Enable (Note 5-24)	SLEEP_FLAG (Note 5-13)	Block SLEEP_EN Signal (Note 5-25)	Description
0	0	0	Block not enabled for sleep, system is in FULL POWER or EC SLEEP state.
	1		Block not enabled for sleep, system is in PREPARING SYSTEM SLEEP or SYSTEM LIGHT SLEEP state.
1	0		Block enabled for sleep, system is in FULL POWER or EC SLEEP state.
	1	1	Block enabled for sleep, system is in PREPARING SYSTEM SLEEP, SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP or SYSTEM DEEPEST SLEEP state.

Note 5-24 Sleep enable for a single block as defined in Section 5.7.5, "Block Sleep Enable Registers," on page 105.

Note 5-25 Clock Generator sleep enable output signal to a single block (SLEEP_EN) as defined in the Generic Block Clocking Model.

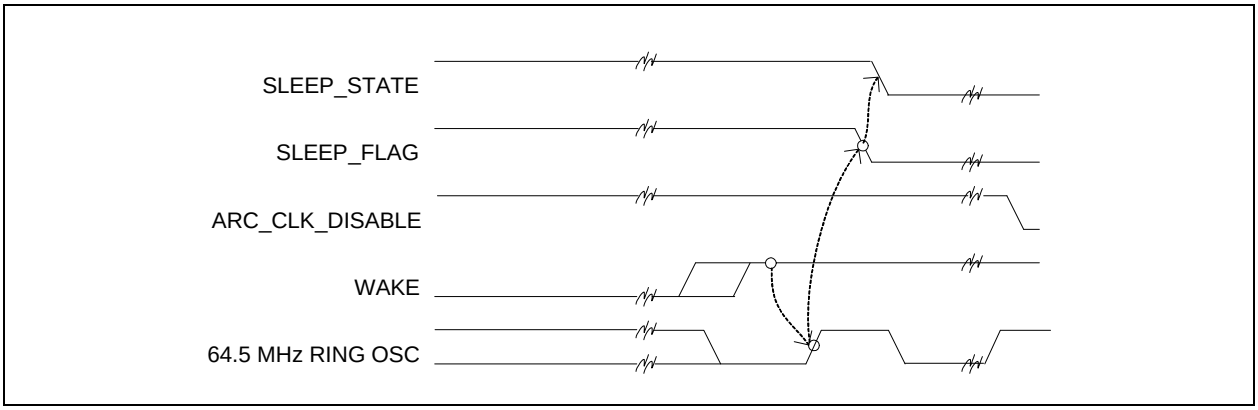
5.4.7.6 Wake Interface

The Wake Interface terminates the sleep states illustrated in Table 5-8, “EC Controlled Dynamic Power States,” on page 841 and includes interrupts to the EC to transition from the EC SLEEP state, as well as wake events that can restart the 64.52 MHz Ring Oscillator and terminate the SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP and SYSTEM DEEPEST SLEEP states.

The WAKE signal shown in Table 5-1, “Power, Clocks and Resets Port List,” on page 74 is the only required input to the Clock Generator for the Wake Interface. Wake Capable Interrupts, although technically part of the Wake Interface, are external to the Power, Clocks and Resets function and are not illustrated in this definition. The minimum pulse-width for wake events is characterized in Note 22-1 on page 335.

When the WAKE signal is asserted while a dynamic system sleep state is enabled, the SLEEP_FLAG in the Clock Control Register is automatically de-asserted by hardware within the first one or two clocks that occur when the 64.52 MHz Ring Oscillator is re-enabled as illustrated in Table 5-8. As described in "EC Power State Controls," on page 87 and in "EC Controlled Sleep State Activation," on page 88, system sleep states are essentially ignored if the SLEEP_STATE signal and the WAKE signal are simultaneously asserted.

FIGURE 5-8: Wake Interface **TIMING**



5.4.7.6.1 Treatment Of Non-wake Capable Interrupts During Sleep Transitions

There are two categories of Interrupts which effect the Wake Interface differently: Wake Capable Interrupts and Non-Wake Capable Interrupts. See Section 16.3.5, "Wake Capable Interrupts," on page 258 in Section 16.0, "EC Interrupt Aggregator".

The lowest power states in which Non-Wake Capable Interrupts can be asserted are PREPARING SYSTEM SLEEP or SYSTEM LIGHT SLEEP. This is because the source of these interrupts require clocks to generate the interrupts. In the event that the EC Controlled Dynamic Power States enters the PREPARING SYSTEM SLEEP or SYSTEM LIGHT SLEEP state by the ARC executing a sleep instruction while a least one non-wake interrupt associated block is active, the EC Controlled Dynamic Power States will not enter a lower state until a non-wake interrupt occurs. The non-wake-able interrupt changes the ARC from a halt state to a running state to process the interrupt. Firmware needs to toggle the state of the SLEEP_FLAG bit in the Clock Control Register. If these bit were Wake-up events, the firmware would not need to de-assert the SLEEP_FLAG bit since hardware will have done so already.

APPLICATION NOTE: if an aggressive sleep policy is implemented and firmware attempts to enter sleep when at least one non-wake interrupt associated block is active, then firmware can treat all interrupt identically (this includes both Wake Capable Interrupts or Non-Wake Capable Interrupts). Firmware can service both Wake Capable Interrupts or Non-Wake Capable Interrupts by attempting to toggle the SLEEP_FLAG in the Clock Control Register first '0' then '1' to prepare to sleep.

APPLICATION NOTE: if a less aggressive sleep policy is implemented and firmware never attempts to enter sleep when any non-wake interrupt associated block is active, then firmware can threat all Wake Capable Interrupts identically. Firmware can service all Wake Capable Interrupts by attempting setting the SLEEP_FLAG bit to '1' in the Clock Control Register to prepare to sleep.

5.4.8 RING OSCILLATOR SOURCED CLOCKING

5.4.8.1 Overview

Ring Oscillator Sourced Clocking includes all of the Fixed Clock Domain, Host Clock Domain and Programmable Clock Domains that are derived from the 64.52 MHz Ring Oscillator. Ring Oscillator Sourced Clocking remains active as long as the 64.52 MHz Ring Oscillator is running (see Section 5.4.7.4, "64.52 MHz Ring Oscillator Control," on page 85).

All Ring Oscillator Sourced Clocking and 32K Clock Domain clocking is summarized in Table 5-14.

TABLE 5-14: ALL Clock Generator OUTPUT PORTS SUMMARY

Symbol	Type (Clock or Enable)	Frequency	Reference
MCLK	CLOCK	64.52 MHz	Section 5.4.3, "64.52 MHz Ring Oscillator," on page 77
EC_BUS_CLK_EN	ENABLE	Programmable	"EC Bus Clock," on page 92
LPC_BUS_CLK_EN	ENABLE	Programmable	"LPC Bus Clock," on page 92
MCLK_DIV2_EN	ENABLE	32.26 MHz	Section 5.4.8.3, "Fixed Clock Domain," on page 92
MCLK_DIV4_EN	ENABLE	16.13 MHz	
MCLK_DIV8_EN	ENABLE	8.06 MHz	
MCLK_DIV16_EN	ENABLE	4.03 MHz	
MCLK_DIV32_EN	ENABLE	2.02 MHz	
MCLK_DIV64_EN	ENABLE	1.01 MHz	
MCLK_DIV128_EN	ENABLE	504 KHz	
MCLK_DIV640_EN	ENABLE	101 KHz	
MCLK_DIV64_EN_HST	ENABLE	1.01 MHz	Section 5.4.8.4, "Host Clock Domain," on page 92
X32K_CLK	CLOCK	32.768 KHz	Section 5.4.4, "32.768 KHz Crystal Oscillator," on page 78

5.4.8.2 Programmable Clock Domains

5.4.8.2.1 EC Bus Clock

The EC Bus Clock (EC_BUS_CLK_EN) is a programmable clock that is derived from the EC Clock Divider Register as described in Section 5.7.2 on page 102.

5.4.8.2.2 LPC Bus Clock

The LPC Bus Clock (LPC_BUS_CLK_EN) is a programmable clock that is derived from the LPC_AHB Clock Divider Register as described in Section 5.7.3 on page 103.

5.4.8.3 Fixed Clock Domain

The Fixed Clock Domain represents non-programmable clocks that are derived from the 64.52 MHz Ring Oscillator. The Fixed Clock Domain outputs as shown in Table 5-14 are MCLK, MCLK_DIV2_EN, MCLK_DIV4_EN, MCLK_DIV8_EN, MCLK_DIV16_EN, MCLK_DIV32_EN, MCLK_DIV64_EN, MCLK_DIV128_EN, and MCLK_DIV640_EN.

5.4.8.4 Host Clock Domain

The Host Clock Domain includes clocking for the Legacy Port Functions. Host Clock Domain clock gating is controlled by VCC_PWRGD such that when VCC_PWRGD is not asserted (‘0’), clocks in the Host Clock Domain are ‘off.’

When VCC_PWRGD is asserted (‘1’) clocks in the Host Clock Domain are ‘on’ and the Legacy Port Functions may be affected by the EC Controlled Dynamic Power States as described in Section 5.4.7, "Power Management Interface," on page 83.

The Host Clock Domain output as shown in Table 5-14 is MCLK_DIV64_EN_HST.

5.4.9 32K CLOCK DOMAIN

5.4.9.1 Overview

The 32K Clock Domain represents all of the clocking derived from the 32.768 KHz Crystal Oscillator. The output of the 32.768 KHz Crystal Oscillator is synchronized to the 64.52 MHz Ring Oscillator as described below in Section 5.4.9.2, "Synchronization".

The 32K Clock Domain remains active as long as the 32.768 KHz Crystal Oscillator is running (see "32K_EN," on page 112). The blocks driven by the 32K Clock Domain are summarized in Table 5-15 and are not affected by Clock Gating in the Power Management Interface. Typically, blocks driven by the 32K Clock Domain can generate wake events, even when the 64.52 MHz Ring Oscillator is disabled.

5.4.9.2 Synchronization

The 32K Clock Domain X32K_CLK output is synchronized to the 64.52 MHz Ring Oscillator. Synchronization is disabled under two conditions: 1) when the 64.52 MHz Ring Oscillator is disabled (e.g., in the SYSTEM DEEPEST SLEEP state or during a test mode) and 2) when VTRGD is not asserted (see Section 5.6.2, "VTRGD," on page 96).

5.4.9.3 Summary

The distribution of the 32K Clock Domain throughout the MEC1609/MEC1609i is illustrated in Table 5-15, below.

TABLE 5-15: 32K Clock Domain DRIVEN BLOCKS

Block Name	Block Cross Reference
Week Timer	Section 21.0, "Week Alarm Interface," on page 324
Watch-Dog Timer	Section 17.0, "Watchdog Timer Interface," on page 293
Hibernation Timer 0	Section 20.0, "Hibernation Timer," on page 320
Hibernation Timer 1	
LED Interface	Section 34.0, "LED Interface," on page 444
Watch-Dog Timer Forced Reset	Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99
64.52 MHz Ring Oscillator	Section 5.4.3, "64.52 MHz Ring Oscillator," on page 77 (Note 5-26).

Note 5-26 Synchronization as described in Section 5.4.9.2 does not apply to the output of the 32.768 KHz Crystal Oscillator that is applied to the 64.52 MHz Ring Oscillator.

5.5 Power Configuration

5.5.1 POWER SUPPLIES AND CLOCKS ACPI CONTEXT

The MEC1609/MEC1609i is influenced by three separate power planes, VBAT, VTR, and VCC, as described in Table 6.15. The VBAT and VTR power planes provide power directly to the MEC1609/MEC1609i through the VBAT and VTR pins shown in Table 5-1, "Power, Clocks and Resets Port List". The MEC1609/MEC1609i senses the VCC power state using the VCC_PWRGD input pin. The VBAT, VTR, and VCC power sequencing requirements are as follows (see also Section 5.5.3, "Power-Up Sequence," on page 94):

1. VCC power can be applied simultaneously with or after VTR power.
2. VTR power can be applied simultaneously with or after VBAT.

The typical relationships of the MEC1609/MEC1609i power supplies to the system power states is shown below in Table 6.15. The distribution of the MEC1609/MEC1609i power supplies to the various functional blocks is illustrated in FIGURE 1-1: MEC1609/MEC1609i Top-level Block Diagram on page 6.

The typical relationships of the MEC1609/MEC1609i clocks to the system power states is shown below in Table 6.16. Descriptions of the various clock domains can be found in Section 5.4, "Clock Generator," on page 76.

TABLE 5-16: TYPICAL MEC1609/MEC1609I POWER SUPPLIES VS. ACPI POWER STATES

	ACPI Power State						
Supply Name	**S0 (FULL ON)**	**S1 (POS)**	**S3 (STR)**	**S4 (STD)**	**S5 (SOFT OFF)**	**G3 (MECH OFF)**	**Description**
VBAT	ON	ON	ON	ON	ON	ON	MEC1609/MEC1609i VBAT Well Supply (assuming a TYPE 2 configuration as described in Section 5.5.2, "Power Supply Configurations," on page 94)
VTR	ON	ON	ON	ON/OFF	ON/OFF	OFF	MEC1609/MEC1609i Suspend Supply. (Note 5-27)
VCC	ON	ON	OFF	OFF	OFF	OFF	MEC1609/MEC1609i Runtime Supply (Note 5-28)

Note 5-27 VTR availability in S4 - S5 may depend, for example, on whether AC power is available.

Note 5-28 The MEC1609/MEC1609i senses the VCC power state using the VCC_PWRGD input pin; i.e., VCC power is not directly applied to this device.

TABLE 5-17: TYPICAL MEC1609/MEC1609I CLOCKS VS. ACPI POWER STATES

	ACPI Power State						
Supply Name	**S0 (FULL ON)**	**S1 (POS)**	**S3 (STR)**	**S4 (STD)**	**S5 (SOFT OFF)**	**G3 (MECH OFF)**	**Description**
32K Clock Domain (XOSEL='1')	ON	ON	ON	ON	ON	OFF/ ON	When XOSEL ='1' and the XTAL2 pin is driven, for example, by an Intel ICH4M SUSCLK, the 32K Clock Domain is running whenever RSMRST is not asserted. (see Ref[6]).
32K Clock Domain (XOSEL='0')	ON	ON	ON	ON	ON	ON	When XOSEL ='0' and a 32.768KHz crystal is connected between the XTAL1 and XTAL2 pins, the 32K Clock Domain is running whenever VBAT is fully powered except following a VBAT_POR as described in "32K_EN," on page 112.

TABLE 5-17: TYPICAL MEC1609/MEC1609I CLOCKS VS. ACPI POWER STATES (CONTINUED)

	ACPI Power State						
Supply Name	**S0 (FULL ON)**	**S1 (POS)**	**S3 (STR)**	**S4 (STD)**	**S5 (SOFT OFF)**	**G3 (MECH OFF)**	**Description**
Host Clock Domain	ON	ON	OFF	OFF	OFF	OFF	The Host Clock Domain is gated by the MEC1609/MEC1609i runtime supply (VCC) as described in Section 5.4.8.4, "Host Clock Domain," on page 92.
PCI_CLK	ON/OFF	ON/OFF	OFF	OFF	OFF	OFF	33MHz LPC Bus clock input powered by the MEC1609/MEC1609i runtime supply (VCC). (Note 5-29)
Programmable Clock Domains and Fixed Clock Domain	ON/OFF	ON/OFF	ON/OFF	OFF/ON	OFF/ON	OFF	These clocks are powered by the MEC1609/MEC1609i suspend supply (VTR) but may start and stop as described in Section 5.4.7, "Power Management Interface," on page 83. (see also Note 5-27)

Note 5-29 The PCI_CLK can start and stop in S0/S1 as defined in Ref[7].

5.5.2 POWER SUPPLY CONFIGURATIONS

There are two acceptable types of MEC1609/MEC1609i power supply configuration that fundamentally differ based on the need for a backup battery connection to VBAT. In both cases VTR is connected to the suspend supply as described in Section 5.5.1, "Power Supplies and Clocks ACPI Context," on page 93.

5.5.2.1 TYPE 1

TYPE 1 configurations do not use a VBAT backup battery connection. There are two VBAT supply configuration options for TYPE 1 power supply configurations: VBAT tied to VSS and VBAT tied to VTR.

In TYPE 1 configurations where VBAT is tied to VSS, the VBAT-Powered Control Interface is unpowered. In this case, the VBAT-powered inputs VCI_IN[3:0]# and VCI_OVRD_IN must be tied to VSS; the VBAT-powered outputs VCI_OUT and BGPO0 must remain unconnected. In TYPE 1 configurations where VBAT is tied to VTR, the VBAT-Powered Control Interface is fully operational when VTR is powered.

Both TYPE 1 configuration options generate a VBAT_POR whenever VTR powers up as described in Section 5.6.3, "VBAT_POR," on page 98.

5.5.2.2 TYPE 2

TYPE 2 configurations use a VBAT backup battery connection (see Table 40-1, “Operating Conditions,” on page 499). Power supply requirements for TYPE 2 configurations are as follows: VBAT is connected to a backup battery that is externally switched with VTR.

In this configuration some internal components that utilize the VBAT power plane are switched internally to VTR using a Power Mux when VTR power is applied as described in Section 5.6.7, "Power Mux," on page 98.

In TYPE 2 configurations, the VBAT-Powered Control Interface can be used to power-on an unpowered system.

5.5.3 POWER-UP SEQUENCE

Table 5-18 summarizes the MEC1609/MEC1609i Power-Up Sequence. For information regarding the typical relationships of the MEC1609/MEC1609i power supplies to the system power states see Section 5.5, "Power Configuration," on page 93.

TABLE 5-18: Power-Up Sequence

	VBAT	VTR	VCC	Reset Interface	Description
1.	OFF	OFF	OFF	–	MEC1609/MEC1609i fully unpowered
2.	ON	OFF	OFF	–	32.768 KHz Crystal Oscillator may be disabled as described in Section 5.6.3, "VBAT_POR," on page 98.
3.	ON	ON	OFF	VBAT_POR, VTRGD, nSYS_RST. nEC_RST, nSIO_RESET	VBAT-powered registers may be reset as described in Section 5.6.3, "VBAT_POR," on page 98. VTR-powered registers and blocks reset (nSYS_RST). EC held in reset as described in Section 5.6.5, "nEC_RST," on page 98.
					nSIO_RESET asserted.
					64.52 MHz Ring Oscillator enabled.
					EC begins code execution following a delay as described in Section 5.6.5, "nEC_RST," on page 98.
4.	ON	ON	ON	VCC_PWRGD, nSIO_RESET	Registers affected by VCC_PWRGD are reset (Note 5-30)
					Firmware de-asserts nSIO_RESET as described in "iRESET OUT," on page 104.

Note 5-30 For VTR-powered on-chip registers that are reset by VCC_PWRGD, it is important that firmware not write to any of these registers until 1 ms following the assertion of VCC_PWRGD (‘1’).

5.6 Reset Interface

5.6.1 OVERVIEW

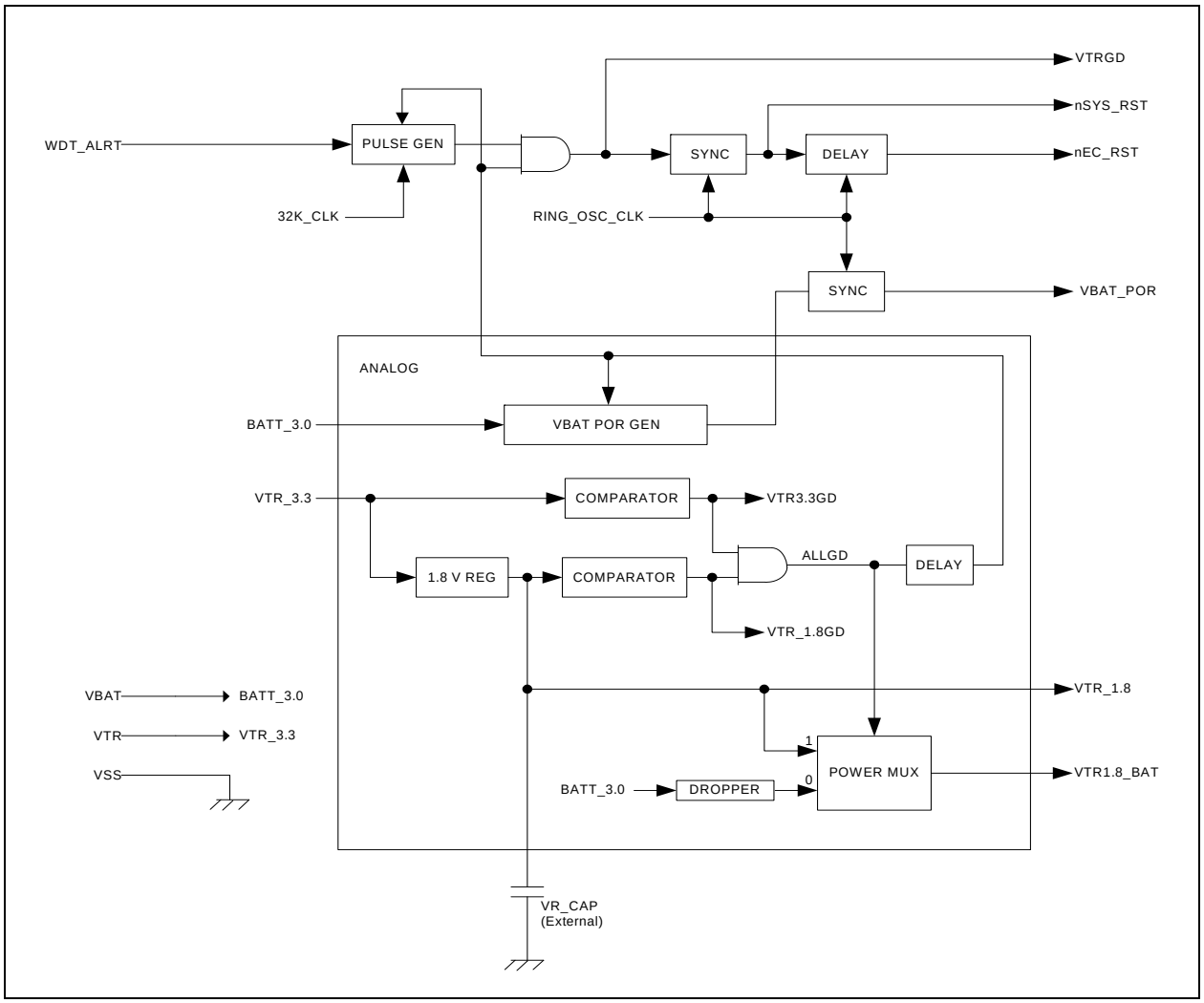
The primary function of the Reset Interface (Figure 5-9) is to generate VTR and VBAT reset signaling; including, VTRGD, VBAT_POR, nSYS_RST, nEC_RST and Watch-Dog Timer Forced Reset (Table 5-19). The Reset Interface also includes a section regarding the 1.8V Regulator. There is other VCC-related Reset Interface functionality not shown in Figure 5-9 that is described Section 5.6.8, "VCC Power Good," on page 99 and Section 5.6.9, "LPC RESET," on page 99.

Also included in the Reset Interface are descriptions of the Power Mux and Registers Interface. These are related Power, Clocks and Resets functions that are not described anywhere else in this chapter.

TABLE 5-19: VTR/VBAT RESET THRESHOLDS

Parameters	MIN	TYP	MAX	Units	Notes
VTRGD Reset Threshold	0.5	1.8	2.7	Volts	
VBAT_POR Reset Threshold	0.5	1.25	1.9		

FIGURE 5-9: Reset Interface BLOCK DIAGRAM



5.6.2 VTRGD

VTRGD is the reset signal for the 64.52 MHz Ring Oscillator and the source for nSYS_RST and nEC_RST.

As shown in Figure 5-9, Figure 5-10 and in Table 5-20, VTRGD is asserted following a delay after the VTR and VTR_1.8 power supplies exceed preset voltage thresholds as defined in Table 5-19, “VTR/VBAT Reset Thresholds,” on page 95. VTRGD is de-asserted as soon as either the VTR or VTR_1.8 power supplies drop below these thresholds (see Figure 5-11, "VTR Power-Down Timing").

VTRGD can also be asserted as a result of a Watch-Dog Timer Forced Reset as described in Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99.

FIGURE 5-10: VTR POWER-UP TIMING

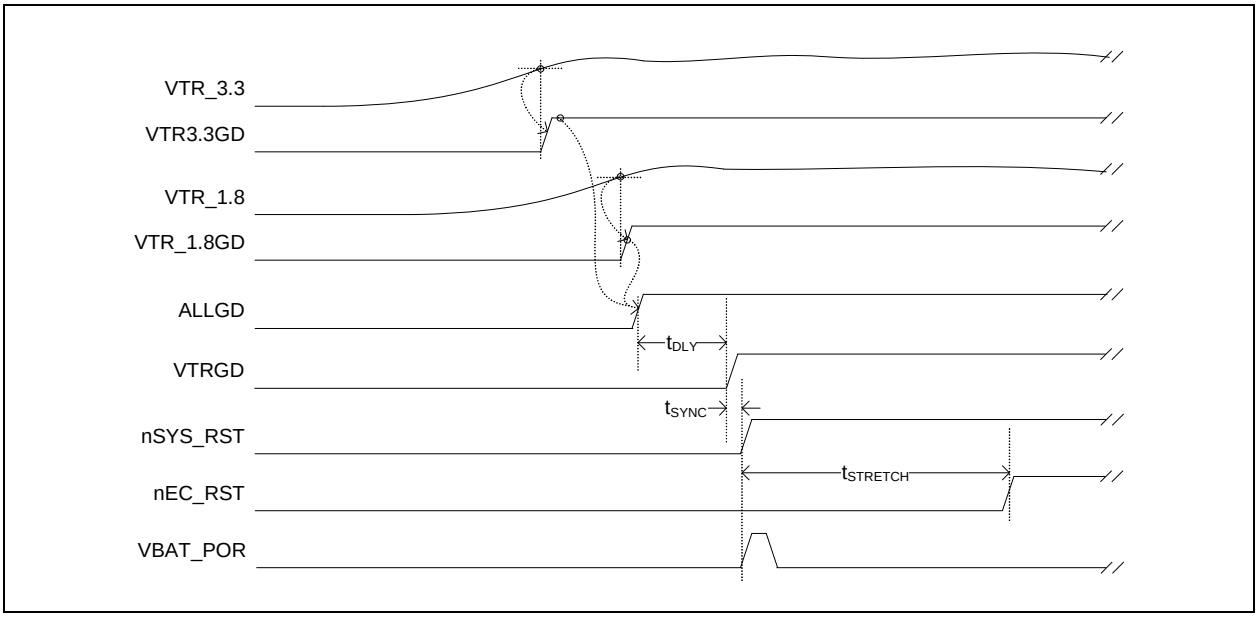
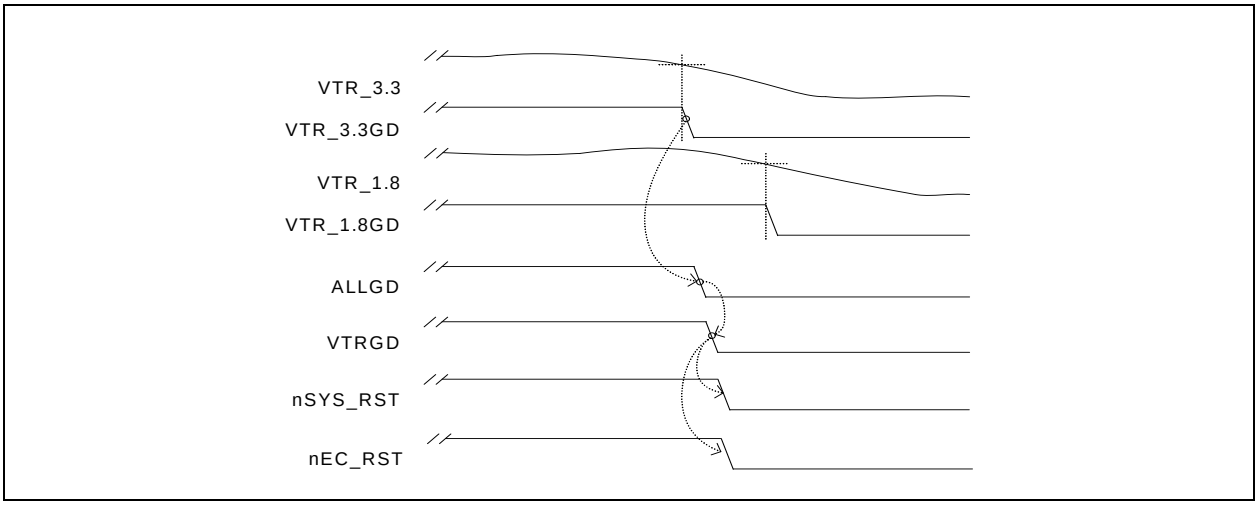


TABLE 5-20: VTR POWER-UP TIMING

Parameters	Symbol	MIN	TYP	MAX	Units	Notes
VTRGD Delay Time	t_{DLY}		360		µs	
nSYS_RST Delay Time	t_{SYNC}	2	–	3	64.52 MHz Ring Oscillator Clocks	Note 5-31
nEC_RST Delay Time	$t_{STRETCH}$	10	20	40	ms	

Note 5-31 This interval is determined using a Fixed Clock Domain from the 64.52 MHz Ring Oscillator.

FIGURE 5-11: VTR POWER-DOWN TIMING



5.6.3 VBAT_POR

VBAT_POR is a pulse that is asserted at the rising edge of nSYS_RST if the coin cell is replaced, or the VBAT voltage falls below 1.25 V nominal (Table 5-19) while nSYS_RST is not asserted (‘0’). Note that the 32.768 KHz Crystal Oscillator is stopped if the coin cell is replaced, or the VBAT voltage falls below 1.25 V nominal while VTRGD is not asserted. No action is taken if the coin cell is replaced, or the VBAT voltage falls below 1.25 V nominal while VTRGD is asserted.

VBAT_POR is used as described throughout this specification to reset registers and functional device blocks. VBAT_POR events are registered in the Power-Fail and Reset Status Register.

5.6.4 NSYS_RST

nSYS_RST is VTRGD synchronized to the 64.52 MHz Ring Oscillator. Note that VTRGD and nSYS_RST have the same logical sense (uninverted); however, because of nomenclature, the asserted states are opposite. Note that VTRGD is defined in Section 5.6.2, "VTRGD," on page 96.

nSYS_RST is de-asserted as defined in Figure 5-10, "VTR Power-Up Timing" and in Table 5-20, "VTR Power-Up Timing". nSYS_RST can also be asserted as a result of a Watch-Dog Timer Forced Reset as described in Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99. nSYS_RST is asserted as soon as either the VTR or VTR_1.8 power supplies drop below preset voltage thresholds (see Figure 6.14 VTR Power-Down Timing on page 117).

nSYS_RST is the reset signal for all VTR-powered blocks except for the 64.52 MHz Ring Oscillator and the Embedded Controller. nSYS_RST also affects the VBAT-Powered Control Interface as described in Table 32-2, “VCI Output Truth Table,” on page 439.

5.6.5 NEC_RST

nEC_RST is a delayed version of nSYS_RST that is used to reset the Embedded Controller and for Registers Interface as described in Section 5.7, "Registers Interface," on page 100.

nEC_RST is de-asserted as defined in Figure 5-10, "VTR Power-Up Timing" and in Table 5-20, "VTR Power-Up Timing". nEC_RST can also be asserted as a result of a Watch-Dog Timer Forced Reset as described in Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99. Like nSYS_RST, nEC_RST is asserted as soon as either the VTR or VTR_1.8 power supplies drop below preset voltage thresholds (see FIGURE 5-11: VTR Power-Down Timing on page 97).

5.6.6 1.8V REGULATOR

The 1.8V Regulator generates the MEC1609/MEC1609i core power well. As illustrated in Figure 5-9, "Reset Interface Block Diagram", the input to the 1.8V Regulator is VTR, the output is VTR_1.8 (see also Table 5-1). The 1.8V Regulator is not used when VTR is inactive, as described in Section 5.6.7, "Power Mux" below.

The stability of the 1.8V Regulator amplifier depends on an external capacitor, VR_CAP as described in Table 5-1. The choice of capacitor can be either ceramic or low ESR tantalum. Ceramics are the recommended choice due to their superior AC performance (below 100mΩ ESR), but X5R dielectrics should be used to prevent greater than 20% capacitance variation over temperature and voltage. Low ESR tantalum capacitors will work but care should be taken because the ESR can vary 2x at low temperatures.

5.6.7 POWER MUX

To provide the highest reliability and lowest possible power consumption, the Power Mux switches between the 1.8V Regulator and a level-shifted VBAT voltage to produce the 1.8V internal supply for VBAT-backed logic (VTR1.8_BAT in Table 5-1).

Power Mux switching depends on the voltage level of the 1.8V Regulator and the VTR supply. As illustrated in Figure 5-9, the Power Mux selects the 1.8V Regulator after the VTR and the VTR_1.8 power supplies exceed preset voltage thresholds. The Power Mux selects the VBAT supply as soon as either the VTR or VTR_1.8 power supplies drop below these thresholds (see FIGURE 5-11: VTR Power-Down Timing on page 97).

Note that the Power Mux only switches 1.8 volts. To ensure minimum VBAT power consumption for 3.3V VBAT powered outputs when VTR is fully powered, supply switching from VBAT to VTR must be done externally.

There is a separate power mux not shown in Figure 5-9 to ensure that the 32.768 KHz Crystal Oscillator is powered when VTR is powered for TYPE 1 Power Supply Configurations.

5.6.8 VCC POWER GOOD

VCC Power Good is defined by the VCC_PWRGD input pin (Table 5-1). VCC_PWRGD is also synchronized to the 64.52 MHz Ring Oscillator and used for the functions shown in Table 5-21.

The VCC_PWRGD input must always be driven to a '1' or a '0,' even when VCC is 0 V. The minimum VCC_PWRGD pulse width (high and low) is shown below in Table 5-21.

TABLE 5-21: VCC_PWRGD INPUT TIMING

Parameters	Symbol	MIN	TYP	MAX	Units	Notes
VCC_PWRGD Pulse Width	t_{VPGPW}	31	–	–	ns	

TABLE 5-22: FUNCTIONS AFFECTED BY VCC Power Good

Name	Reference
Host Clock Domain	see Section 5.4.8.4, "Host Clock Domain," on page 92
LPC RESET	Section 5.6.9, "LPC RESET," on page 99
nSIO_RESET	"iRESET OUT," on page 104.
VCC_PWRGD_BUFF	Table 5-1, "Power, Clocks and Resets Port List," on page 74.
VCC_PWRGD	"VCC PWRGD," on page 104

5.6.9 LPC RESET

LPC RESET (LPC_RST# in Table 5-1) is defined by VCC_PWRGD**,** LRESET# and VTRGD as illustrated in Table 5-23. LPC RESET only affects logic that is driven by PCI_CLK.

TABLE 5-23: LPC RESET DEFINITION

VCC_PWRGD (Note 5-32)	LRESET# (Note 5-33)	VTRGD (Note 5-34)	LPC RESET (Note 5-35)
0	X	X	0
1		0	Undefined
	0	1	0
	1		1

Note 5-32 This is the Table 5-1 VCC_PWRGD input.

Note 5-33 This is the Table 5-1 LRESET# input. The EC can determine the state of the LRESET# input using registers in Table 6-8, "LPC Bus Monitor Register," on page 127.

Note 5-34 See Section 5.6.2, "VTRGD," on page 96.

Note 5-35 LPC RESET is the Table 5-1 LPC_RST# output. The trailing edge of LPC_RST# is synchronized to the PCI_CLK in Table 5-1.

5.6.10 WATCH-DOG TIMER FORCED RESET

A Watch-Dog Timer Forced Reset (Figure 5-12) occurs when the WDT_ALRT input (Table 5-1) is asserted ('1'). As shown in Figure 5-12, VTRGD is de-asserted ('0') and nSYS_RST and nEC_RST are asserted ('0') when WDT_ALRT is asserted. The VTRGD reset time (t_{RST}) is determined by the 32.768 KHz Crystal Oscillator as shown in Table 5-24. Following the VTRGD reset time, the nSYS_RST Delay Time (t_{SYNC}) and the nEC_RST Delay Time ($t_{STRETCH}$) are determined using the 64.52 MHz Ring Oscillator as described in Figure 5-10, "VTR Power-Up Timing" and Table 5-20, "VTR Power-Up Timing".

Note that analog reset signal functions are not shown in Figure 5-12 because it is assumed that the power supplies are fully powered and stable during a Watch-Dog Timer Forced Reset.

FIGURE 5-12: Watch-Dog Timer Forced Reset TIMING

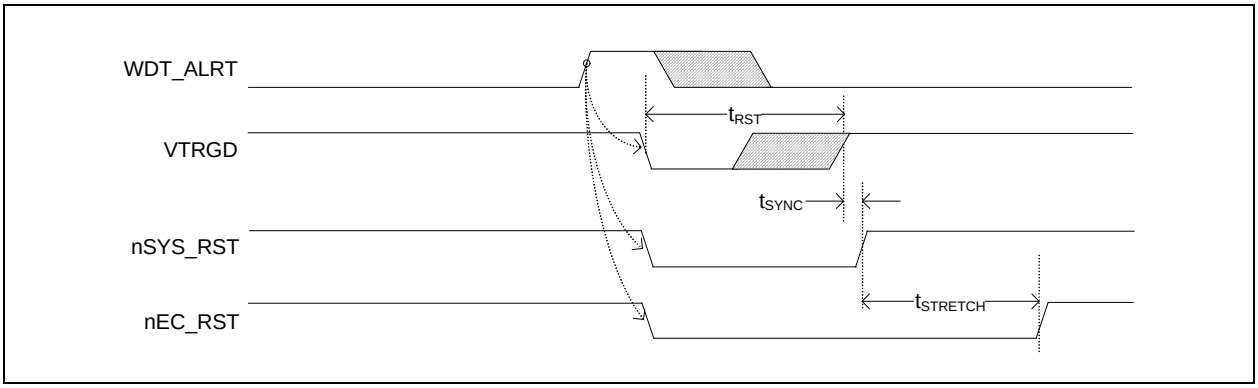


TABLE 5-24: Watch-Dog Timer Forced Reset TIMING

Parameters	Symbol	MIN	TYP	MAX	Units
VTRGD Reset Time	t_{RST}	1	–	2	32.768 KHz Crystal Oscillator Clock Cycles
nSYS_RST Delay Time	t_{SYNC}	(see FIGURE 5-10: VTR Power-Up Timing on page 97			
nEC_RST Delay Time	$t_{STRETCH}$				

5.7 Registers Interface

The Power, Clocks and Resets registers are located in two address ranges with two Base Address as indicated in Table 5-25. See Note 3-1 on page 49.

TABLE 5-25: POWER, CLOCKS AND RESETS INTERFACE BASE ADDRESS TABLE

Power, Clocks and Resets Blocks	LDN from (Table 3-2 on page 48)	AHB Base Address
Power, Clock & Reset (VTR PWR'ed)	32h	F0_C800h
Power, Clock & Reset (VBAT PWR'ed)	33h	F0_CC00h

Table 5-26 is a register summary for the Power, Clock & Reset (VTR PWR'ed) registers. Table 5-27 is a register summary for the Power, Clock & Reset (VBAT PWR'ed) registers.

Each EC address is indicated as an SPB Offset from its AHB base address as indicated in Table 5-25.

The following tables summarize the registers allocated for each Instance. The offset field in the following table is the offset from the Embedded Controller (EC) Base Address.

TABLE 5-26: Power, Clocks and Resets VTR-POWERED REGISTERS SUMMARY

Offset (HEX) (Note 5-36)	Register Name	Access	Size (Bits)	Page Reference
0	EC Clock Divider Register	R/W	4	102
4	LPC_AHB Clock Divider Register	R/W	8	124
8	PCR Status and Control Register	R	8	104
C	Clock Control Register	R/W	8	101
10	LPC Blocks Sleep Enables Register	R/W	8	105
14	EC Blocks Sleep Enables Register 1	R/W	24	106
18	LPC Blocks Clock Required Status Register	R	8	108

TABLE 5-26: Power, Clocks and Resets VTR-POWERED REGISTERS SUMMARY (CONTINUED)

Offset (HEX) (Note 5-36)	Register Name	Access	Size (Bits)	Page Reference
1C	EC Blocks Clock Required Status Register 1	R	24	106
20	OSC_ID Register	R	8	110
24	Reserved	R	32	–
30h	EC Blocks Sleep Enables Register 2	R/W	16	106
34h	EC Blocks Clock Required Status Register 2	R	16	109

Note 5-36 All register addresses are naturally aligned on 32-bit boundaries. Offsets for registers that are smaller than 32 bits are reserved and must not be used for any other purpose.

TABLE 5-27: Power, Clocks and Resets VBAT-POWERED REGISTERS SUMMARY

Offset (HEX) (Note 5-36)	Register Name	Access	Size (Bits)	Page Reference
0	Power-Fail and Reset Status Register	R/W	8	111
4	Clock Enable Register	R/W	8	111

5.7.1 CLOCK CONTROL REGISTER

TABLE 5-28: Clock Control Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	0Ch			8-bit			**EC SIZE**	
POWER	VTR			04h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R/W	R/W	R/W	R/W	R
BIT NAME	Reserved			SAA	32KHz OUTPUT	ROSC_SLP_OVRD	SLEEP_FLAG	Reserved

SLEEP_FLAG

The SLEEP_FLAG affects the system power state as described in Section 5.4.7.2, "EC Controlled Dynamic Power States," on page 84. The SLEEP_FLAG is R/W. EC firmware asserts SLEEP_FLAG (‘1’), which is then typically de-asserted (‘0’) by hardware as described in Section 5.4.7.6, "Wake Interface," on page 90.

ROSC_SLP_OVRD

The EC can prevent the start-up time of the 64.52 MHz Ring Oscillator from affecting system sleep state latency by using the 64.52 MHz Ring Oscillator Sleep Override bit (ROSC_SLP_OVRD). As illustrated in Figure 5-5, "64.52 MHz Ring Oscillator Controls", during normal operation when ROSC_SLP_OVRD is asserted (‘1’ default), the 64.52 MHz Ring Oscillator will always remain enabled. When ROSC_SLP_OVRD is not asserted (‘0’), the 64.52 MHz Ring Oscillator can be stopped as described in Section 5.4.7.4, "64.52 MHz Ring Oscillator Control," on page 85.

32KHZ OUTPUT

The 32KHz OUTPUT bit controls the MEC1609/MEC1609i 32KHZ_OUT signal function (Table 5-1). When 32KHz OUTPUT is de-asserted ‘0,’ the 32KHz output clock is disabled and the 32KHZ_OUT signal function is driven low. When 32KHz OUTPUT is asserted, the 32KHz output clock is enabled. The 32KHz OUTPUT bit is R/W and disabled by default following VTRGD.

SAA

When asserted (‘1’), the Stop Auto-Adjust bit (SAA) disables automatic frequency correction of the 64.52 MHz Ring Oscillator. When SAA is not asserted (‘0’) (default), the 64.52 MHz Ring Oscillator operates normally as defined in Section 5.4.3, "64.52 MHz Ring Oscillator," on page 77.

Note that when SAA is asserted voltage and temperature variations can adversely affect the frequency of the 64.52 MHz Ring Oscillator. To provide the accuracy of the 64.52 MHz Ring Oscillator as defined in Section 5.4.3, "64.52 MHz Ring Oscillator", the SAA bit must not be asserted.

APPLICATION NOTE: note that the FREQ LOCK function as described in "FREQ LOCK," on page 104 is undefined when the SAA bit is asserted.

5.7.2 EC CLOCK DIVIDER REGISTER

5.7.2.1 Overview

The EC Clock Divider Register (Table 5-30) contains the EC_CLK_DIV bits that are used to program the EC clock and the EC_AHB clock enable frequency as described in "EC_CLK_DIV," on page 102.

In the MEC1609/MEC1609i, the highest available frequency that can be programmed using the EC Clock Divider Register is 32.25 MHz (EC_CLK_DIV = 2), the lowest is 4.3 MHz (EC_CLK_DIV = 15). As shown in Table 5-29, normal operation is only maintained for EC_CLK_DIV values 03h - 0Fh. See Section 15.3, "EC Clocking," on page 245 for a detailed description of EC clocking.

TABLE 5-29: EC_CLK_DIV PROGRAMMING BEHAVIOR

EC_CLK_DIV	Frequency (MHz)	Flash Access Allowed	Description
0h - 1h	NO CHANGE	–	Reserved. Writes to the EC Clock Divider Register are ignored.
2h	32.25	NO	Only accesses to AHB registers or 32-bit aligned-code accesses to SRAM. See Section 15.3, "EC Clocking," on page 245.
3h - Fh	21.5 – 4.3	YES	Normal Operation.

TABLE 5-30: EC Clock Divider Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	00h				8-bit		**EC SIZE**	
POWER	VTR				08h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EC_CLK_DIV			

EC_CLK_DIV

The EC_CLK_DIV bits contain the binary encoded divider that determines the EC clock and EC_AHB clock enable frequency. Valid EC_CLK_DIV values are 2h - Fh. The EC_CLK_DIV default is 08h (8 MHz). Writing a ‘0h’ or a ‘1h’ to the EC Clock Divider Register has no affect.

When the EC_CLK_DIV is greater than ‘01h,’ the EC clock and EC_AHB clock enable frequency (F) is calculated using the equation in Figure 5-13, where MCLK is the undivided output of the 64.52 MHz Ring Oscillator and DIV is the EC_CLK_DIV value programmed into the EC Clock Divider Register.

FIGURE 5-13: EC_CLK_DIV EQUATION

$$F = \frac{MCLK}{DIV}$$

APPLICATION NOTE: to support EC traffic to the LPC Subsystem, the EC_AHB clock frequency must be equal to or less than the LPC_AHB clock frequency.

APPLICATION NOTE: the JTAG clock can't be higher than 1/2 the EC clock as defined by the EC Clock Divider Register.

5.7.3 LPC_AHB CLOCK DIVIDER REGISTER

5.7.3.1 Overview

The LPC_AHB Clock Divider Register contains the LPC_AHB_CLK_DIV bits that are used to program the LPC_AHB clock enable frequency as described in Section "LPC_AHB_CLK_DIV," on page 125.

In the MEC1609/MEC1609i, the highest available frequency that can be programmed using the LPC_AHB Clock Divider Register is 64.52 MHz (LPC_AHB_CLK_DIV = 1), the lowest is 4.3 MHz (LPC_AHB_CLK_DIV = Fh). As shown in Table 5-31, normal operation is only provided for LPC_AHB_CLK_DIV values 01h - 0Fh.

TABLE 5-31: LPC_AHB_CLK_DIV PROGRAMMING BEHAVIOR

LPC_AHB_CLK_DIV	Frequency (MHz)	Description
00h	NO CHANGE	Reserved. Writes to the LPC_AHB Clock Divider Register are ignored.
01h - 0Fh	64.52 – 4.30	Normal Operation.

TABLE 5-32: LPC_AHB Clock Divider Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	04h				8-bit		**EC SIZE**	
POWER	VTR				01h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				LPC_AHB_CLK_DIV			

LPC_AHB_CLK_DIV

The LPC_AHB_CLK_DIV is the binary encoded divider that determines the LPC_AHB clock enable frequency. Valid LPC_AHB_CLK_DIV values are 01h - 0Fh. The LPC_AHB_CLK_DIV default is 01h. Writing a '00h' to the LPC_AHB Clock Divider Register has no affect.

When the LPC_AHB_CLK_DIV is greater than '00h,' the LPC_AHB clock enable frequency (F) is calculated using the same equation as for the EC_CLK_DIV (Figure 5-13), where MCLK is the undivided output of the 64.52 MHz Ring Oscillator and DIV is the LPC_AHB_CLK_DIV value. When the LPC_AHB_CLK_DIV is '01h,' the LPC_AHB clock frequency (F) is the undivided output of the 64.52 MHz Ring Oscillator.

APPLICATION NOTE: to support EC traffic to the LPC Subsystem, the EC_AHB clock frequency must be equal to or less than the LPC_AHB clock frequency.

5.7.4 PCR STATUS AND CONTROL REGISTER

TABLE 5-33: PCR Status and Control Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	08h			8-bit			**EC SIZE**	
POWER	VTR			00XX_XX1Xb			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R/W	R	R	R	R
BIT NAME	Reserved		FREQ LOCK	iRESET OUT	Reserved	Reserved	VCC PWRGD	Reserved

VCC PWRGD

The VCC PWRGD bit reflects the state of the synchronized VCC_PWRGD input pin (see Section 5.6.8, "VCC Power Good," on page 99). The VCC PWRGD pin can generate an either-edge interrupt as described in the note associated with the VCC PWRGD signal in Section 22.4, "GPIO Indexing," on page 330.

IRESET OUT

The iRESET OUT bit is used by firmware to control the nSIO_RESET signal function (Table 5-1). Firmware can program the state of iRESET OUT except when the VCC PWRGD bit is not asserted ('0'), in which case iRESET OUT is 'don't care' and nSIO_RESET is asserted ('0') (Table 5-34). In all other cases, the nSIO_RESET signal function is always the inverse of the iRESET OUT bit.

APPLICATION NOTE: it should be noted that when the iRESET OUT bit is asserted ('1') the internal nSIO_RESET is asserted even if the nRESET_OUT pin is configured as an alternate function.

Note 5-37 When the iRESET OUT bit is set to '1', the falling edge of VCC PWRGD will cause the nRESET_OUT signal function to be asserted within 15ns, A subsequent rising edge of VCC PWRGD will cause the nRESET_OUT signal function to de-assert within 3 MCLKs.

Note 5-38 nSIO_RESET is also the source for the nRESET_OUT signal function (Table 2-15, "Miscellaneous Functions," on page 17).

TABLE 5-34: iRESET OUT **BIT BEHAVIOR**

VCC PWRGD	iRESET OUT	nSIO_RESET **& nRESET_OUT (See Note 5-38)**	**Description**
0	X	0 (ASSERTED)	The iRESET OUT bit does not affect the state of nSIO_RESET when VCC PWRGD is not asserted.
1	1	0 (ASSERTED)	The iRESET OUT bit can only be written by firmware when VCC PWRGD is asserted.
	0	1 (NOT ASSERTED)	

FREQ LOCK

FREQ LOCK is asserted ('1') when the 32.768 KHz Crystal Oscillator is stable and the accuracy of the 64.52 MHz Ring Oscillator is within the tightest tolerance described in Table 5-3, "64.52 MHz Ring Oscillator Timing Parameters," on page 77.

5.7.5 BLOCK SLEEP ENABLE REGISTERS

The Block Sleep Enables identified in the Block Sleep Enable Registers are described below in Section 5.7.5.1, "LPC Blocks Sleep Enables Register," on page 105 and Section 5.7.5.2, "EC Blocks Sleep Enables Registers," on page 106. The behavior of the Block Sleep Enables is described in "Block Sleep Enables," on page 89.

5.7.5.1 LPC Blocks Sleep Enables Register

TABLE 5-35: LPC Blocks Sleep Enables Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	10h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	MCHP Reserved	EMI	FLASH SPI	MCHP Reserved	UART	MCHP Reserved	VLPC

TABLE 5-36: LPC BLOCKS SLEEP ENABLES/CLOCK REQUIRED REGISTERS BIT NAMES

Bit Name	Block Name	Block Cross Reference
VLPC	VLPC Interface	Section 8.0, "VLPC Bus Interface," on page 153
UART	16C550A UART	Section 13.8, "Sleep Enable/ Clock Request Power state controls," on page 218
FLASH SPI	LPC GP-SPI	Section 31.0, "General Purpose Serial Peripheral Interface (GP-SPI)," on page 413
EMI	Embedded Memory Interface	Section 7.0, "Embedded Memory Interface," on page 134

Some LPC accessible blocks have no Block Sleep Enable bit in the LPC Blocks Sleep Enables Register. Table 5-37 describes these.

TABLE 5-37: LPC BLOCKS NOT CONTROLLED BY LPC Blocks Sleep Enables Register

Block Name	Block Cross Reference
LPC Interface	Section 6.10.4, "EC Clock Control Register," on page 130
Legacy Port Functions	Legacy Support on page 178
ACPI EC Interface	Section 9.0, "ACPI Embedded Controller Interface," on page 162
8042 Emulated Keyboard Controller Interface	Section 10.0, "8042 Emulated Keyboard Controller," on page 170
ACPI PM1 Interface	Section 11.0, "ACPI PM1 Block Interface," on page 185
Mailbox Registers Interface	Section 12.0, "MailBox Register Interface," on page 193

5.7.5.2 EC Blocks Sleep Enables Registers

TABLE 5-38: EC BLOCKS SLEEP ENABLES REGISTER 1

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	14h				24-bit		**EC SIZE**	
POWER	VTR				00_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	SPI_P	SMB1	SMB0	PWM3	PWM2	PWM1	PWM0
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R	R	R	R	R/W
BIT NAME	PS2_2	PS2_1	PS2_0	Reserved				TACH2
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACH1	TACH0	Reserved	RC_ID	C/T3	C/T2	C/T1	C/T0

TABLE 5-39: EC BLOCKS SLEEP ENABLES REGISTER 2

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	30h				16-bit		**EC SIZE**	
POWER	VTR				0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				SPI Flash Read	KSC	PECI	ADC
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMA	CCT	SMB2	TACH3	PWM7	PWM6	PWM5	PWM4

TABLE 5-40: EC BLOCKS SLEEP ENABLES/CLOCK REQUIRED REGISTERS BIT NAMES

Bit Name	Block Name (Note 5-39)	Block Cross Reference
C/T0	16-Bit Counter/Timer 0	Section 19.0, "16-Bit Timer Interface," on page 305
C/T1	16-Bit Counter/Timer 1	
C/T2	16-Bit Counter/Timer 2	
C/T3	16-Bit Counter/Timer 3	
RC_ID	RC ID Interface	Section 30.0, "RC Identification Detection (RC_ID)," on page 403
Reserved	–	–
TACH0	Tachometer 0	Section 28.0, "TACH Monitor," on page 388
TACH1	Tachometer 1	
TACH2	Tachometer 2	
TACH3	Tachometer 3	
PS2_0	PS/2 Interface 0	Section 35.0, "PS/2 Device Interface," on page 451
PS2_1	PS/2 Interface 1	
PS2_2	PS/2 Interface 2	
PWM0	PWM 0	Section 29.0, "PWM Controller," on page 397l
PWM1	PWM 1	
PWM2	PWM 2	
PWM3	PWM 3	
PWM4	PWM 4	
PWM5	PWM 5	
PWM6	PWM 6	
PWM7	PWM 7	
SMB0	SMBus 0	Section 25.0, "SMB Device Interface," on page 372
SMB1	SMBus 1	
SMB2	SMBus 2	
SPI_P	SPI Peripheral Interface	Section 31.0, "General Purpose Serial Peripheral Interface (GP-SPI)," on page 413
CCT	Capture Compare Timer	Section 23.0, "Input Capture and Compare Timer," on page 347
DMA	DMA	Section 24.0, "DMA Controller," on page 360
ADC	ADC	Section 27.0, "Analog to Digital Converter (ADC)," on page 378
PECI	PECI	Section 26.0, "PECI Interface," on page 375
KSC	Key scan	Section 36.0, "Keyboard Matrix Scan Support," on page 459
SPI Flash Read	AHB SPI Flash Read	Section 18.0, "EC AHB SPI Flash Read Controller," on page 299
Reserved	–	–

Note 5-39 Some EC accessible blocks have no Block Sleep Enable bit in the EC Blocks Sleep Enables Register 1. Table 5-41 describes these.

TABLE 5-41: EC BLOCKS NOT CONTROLLED BY EC Blocks Sleep Enables Registers

Block Name	Block Cross Reference
MCU Serial Debug Port	Section 38.0, "Serial Debug Port," on page 473
Master BC Link D	Section 37.0, "BC-Link Master," on page 465
Master BC Link A	
Master BC Link B	
Master BC Link C	
Flash Interface	Section 14.0, "Embedded Flash Subsystem," on page 219
EC	Section 15.0, "ARC 625D Embedded Controller," on page 244
Interrupt Aggregater	Section 16.0, "EC Interrupt Aggregator," on page 251 (see also Section 5.4.6.2.2, "INTERRUPT AGGREGATOR CLOCK TREE," on page 81)
Hibernation Timer	Section 20.0, "Hibernation Timer," on page 320
Week Alarm Timer	Section 21.0, "Week Alarm Interface," on page 324
GPIO	Section 22.0, "GPIO Interface," on page 329 (see also Section 5.4.6.2.3, "GPIO CLOCK TREE," on page 81)
VCI	Section 32.0, "VBAT-Powered Control Interface," on page 437
VBAT_RAM	Section 33.0, "VBAT Powered RAM," on page 442
LED	Section 34.0, "LED Interface," on page 444

5.7.6 CLOCK REQUIRED STATUS REGISTERS

The Clock Required Status Registers indicates the core clock status per block as defined in Section 5.4.5, "Generic Block Clocking Model," on page 78. Like the Block Sleep Enable Registers, there are two types of Clock Required Status Registers: the LPC Blocks Clock Required Status Register and the EC Blocks Clock Required Status Registers.

When a bit in the Clock Required Status Registers is asserted ('1'), the block is enabled and requires that the 64.52 MHz Ring Oscillator remain running as defined in "EC Power State Controls," on page 87.

When a bit in the Clock Required Status Registers is not asserted ('0'), the block is either not enabled as defined in the Generic Block Clocking Model, or has been commanded to sleep and no longer requires the 64.52 MHz Ring Oscillator.

5.7.6.1 LPC Blocks Clock Required Status Register

TABLE 5-42: LPC Blocks Clock Required Status Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	18h				8-bit		**EC SIZE**	
POWER	VTR				0Xh		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved	MCHP Reserved	EMI	FLASH SPI	LEGACY	UART	LPC	VLPC

5.7.6.2 EC Blocks Clock Required Status Registers

TABLE 5-43: EC BLOCKS CLOCK REQUIRED STATUS REGISTER 1

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	1Ch				24-bit		**EC SIZE**	
POWER	VTR				XX_XXXXh		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	FLASH Note 5-40	SPI_P	SMB1	SMB0	PWM3	PWM2	PWM1	PWM0
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	PS2_2	PS2_1	PS2_0	MBCLC	MBCLB	MBCLA	MBCLD	TACH2
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	TACH1	TACH0	MSDP	RC_ID	C/T3	C/T2	C/T1	C/T0

Note 5-40 The Flash Clock Required status bit FLASH is asserted whenever the ARC or a JTAG Debug master attempts to access the EC Blocks Clock Required Status Registers.

TABLE 5-44: EC BLOCKS CLOCK REQUIRED STATUS REGISTER 2

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	34h				16-bit		**EC SIZE**	
POWER	VTR				0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved				SPI Flash Read	KSC	PECI	ADC
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	DMA	CCT	SMB2	TACH3	PWM7	PWM6	PWM5	PWM4

TABLE 5-45: BLOCKS NOT CONTROLLED BY SLEEP ENABLE BITS WITH CLOCK REQUIRED STATUS OUTPUTS

Bit Name	Block Name	Block Cross Reference
LEGACY	Legacy Port Functions	Legacy Support on page 178
UART	16C550A UART	Section 13.8, "Sleep Enable/ Clock Request Power state controls," on page 218
LPC	LPC Interface	Section 6.10.4, "EC Clock Control Register," on page 130
FLASH	Flash Interface	Section 14.0, "Embedded Flash Subsystem," on page 219
MBCLD	Master BC Link D	Section 37.0, "BC-Link Master," on page 465
MBCLA	Master BC Link A	
MBCLB	Master BC Link B	
MBCLC	Master BC Link C	
MSDP	MCU Serial Debug Port	Section 38.0, "Serial Debug Port," on page 473

5.7.7 OSC_ID REGISTER

The OSC_ID Register (Table 6.41) contains the FOUNDRY, FOUNDRY and SHRINK identification codes for the 64.52 MHz Ring Oscillator.

TABLE 5-46: OSC_ID Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	20h			8-bit			**EC SIZE**	
POWER	VTR			XXh			**HARDWIRED DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	BLOCK_REVISION				FOUNDRY		SHRINK	

SHRINK

The 2-bit SHRINK register represents the hard-coded 64.52 MHz Ring Oscillator shrink factor.

FOUNDRY

The 2-bit FOUNDRY register represents the hard-coded 64.52 MHz Ring Oscillator foundry code.

BLOCK_REVISION

The 4-bit BLOCK_REVISION register represents the hard-coded 64.52 MHz Ring Oscillator block revision number.

5.8 VBAT Powered Registers

5.8.1 POWER-FAIL AND RESET STATUS REGISTER

5.8.1.1 Overview

The Power-Fail and Reset Status Register collects and retains the VBAT RST, FLASH and WDT event status when VTR is unpowered. Asserted events can cause interrupts as described in Section 5.9, "Interrupt Interface," on page 112.

TABLE 5-47: Power-Fail and Reset Status Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	00h			8-bit			**EC SIZE**	
POWER	VBAT			1XXX000b			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/WC	R/WC	R/WC	R	R	R	R	R
BIT NAME	VBAT RST	FLASH	WDT	Reserved				

WDT

The WDT bit is asserted (‘1’) following a Watch-Dog Timer Forced Reset as described in Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99. To clear the WDT bit EC firmware must write a ‘1’ to this bit; writing a ‘0’ to the WDT bit has no affect.

FLASH

The FLASH bit is set to ‘1’ by hardware when FLASH_PGM in Table 5-1, “Power, Clocks and Resets Port List,” on page 74 is asserted. FLASH_PGM is asserted when the Embedded Flash Subsystem is placed in Program Mode or Erase Mode. To clear the FLASH bit EC firmware must write a ‘1’ to this bit; writing a ‘0’ to the FLASH bit has no affect.

VBAT RST

The VBAT RST bit is set to ‘1’ by hardware when a VBAT_POR is detected. This is the register default value. To clear VBAT RST EC firmware must write a ‘1’ to this bit; writing a ‘0’ to VBAT RST has no affect.

5.8.2 CLOCK ENABLE REGISTER

TABLE 5-48: Clock Enable Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VBAT			00h			VBAT_POR **DEFAULT**	
				N/A			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						32K_EN	XOSEL

XOSEL

When the External Oscillator Select bit (XOSEL) is asserted ('1'), the 32.768 KHz Crystal Oscillator is driven by a single-ended 32.768 KHz clock source connected to the XTAL2 pin; the XTAL1 pin must be grounded.

When XOSEL is not asserted ('0') (default), the 32.768 KHz Crystal Oscillator requires a 32.768 KHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins. See also Table 5-17, "Typical MEC1609/MEC1609i Clocks vs. ACPI Power States," on page 93.

APPLICATION NOTE: The XOSEL bit should be correctly configured by firmware before the 32K_EN bit is asserted.

32K_EN

The 32K_EN bit controls the 32.768 KHz Crystal Oscillator as defined in Table 5-49. The 32K_EN bit is de-asserted ('0') by default on a VBAT_POR (see also Section 5.6.3 on page 98). The 32K_EN bit is unaffected by nSYS_RST.

TABLE 5-49: 32K_EN BIT

32K_EN	32.768 KHz Crystal Oscillator	Description
0	OFF	VBAT_POR default.
1	ON	The 32.768 KHz Crystal Oscillator can only be enabled by firmware.

5.9 Interrupt Interface

The Power, Clocks and Resets Interrupt Interface inputs include the VBAT RST, FLASH and WDT status bits in the Power-Fail and Reset Status Register. The Interrupt Interface output is PCR_INT in Table 5-1, "Power, Clocks and Resets Port List," on page 74. The corresponding bit in the EC Interrupt Aggregator is bit PFR in GIRQ23 Source Register.

Whenever any Interrupt Interface input is asserted, PCR_INT is asserted; when all Interrupt Interface inputs are not asserted, PCR_INT is not asserted. PCR_INT may be masked as described in Section 16.0, "EC Interrupt Aggregator," on page 251.

6.0 HOST INTERFACE

6.1 General Description

6.1.1 OVERVIEW

The host processor communicates with the MEC1609/MEC1609i via the LPC Bus Interface. The host processor communicates through a series of read/write registers in the MEC1609/MEC1609i. Register access is accomplished through programmed I/O or DMA LPC transfer cycles. All I/O transfer cycles are 8 bits wide. DMA transfer cycles can be 16-bit or 8-bit wide.

The Logical Devices physically located in the MEC1609/MEC1609i are identified in Table 3-2, “Host Logical Devices on MEC1609/MEC1609i,” on page 48 and Table 4-1, “Basechip Logical Devices,” on page 53. The base addresses of logical devices with registers located in LPC I/O space, including the Keyboard Controller, can be moved via the configuration registers located in the LPC Interface Configuration Register Space.

The Logical Devices physically located on companion chips (Companion Logical Devices) are identified in the Companion’s datasheet. LPC I/O & DMA cycles targeting CLD’s are forwarded to the VLPC-Port. The base addresses CLD’s can be programmed and activated via the configuration registers located in the VLPC Configuration Register Space (TWB-Map).

All configuration register access for the MEC1609/MEC1609i basechip and associated companions are accessed indirectly through the LPC I/O Configuration Register Port (IOCR-Port.) The default I/O address is 2Eh and 2Fh, but the IOCR-Port can be relocated by either the host or the EC. Detailed description of the MEC1609/MEC1609i Configuration Space is in Section 4.0, "Logical Device Configuration," on page 53.

Some Configuration Registers associated with CLD’s are physically located in companion chips. These registers are called Companion Configuration Register (CCR) and are identified in the companion datasheet. These registers are indirectly accessed via the LPC I/O Configuration Register Port (IOCR-Port.) LPC I/O cycles targeting CCR’s are forwarded to the TWB-Port and completed on the TWB and then completed on the LPC. Base Address Registers for all CLD’s are located in the LPC Logical Device.

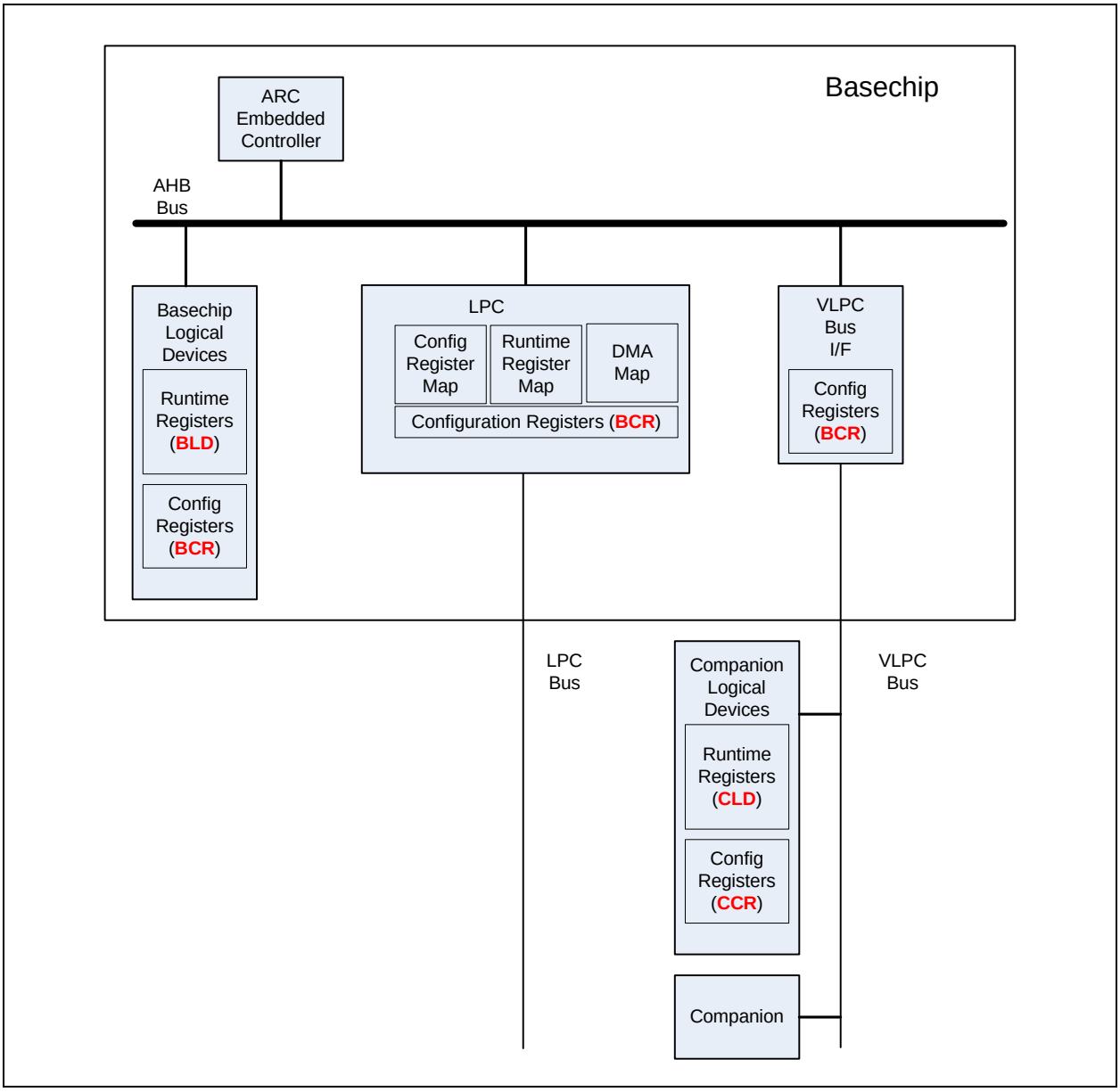
All LPC transactions that are claimed by the MEC1609/MEC1609i are mapped by the LPC interface to an address in the MEC1609/MEC1609i’s AHB address space. All these addresses can also be accessed by the Embedded Controller in the MEC1609/MEC1609i.

TABLE 6-1: TARGETS OF LBC CYCLES CLAIMED BY THE MEC1609/MEC1609I

Target	Acronym	Description	LPC Type
LPC IO Configuration Register Port	**IOCR-Port**	Standard LPC 2Eh/2Fh Port which permits BIOS access. This port can be relocated by the ARC.	I/O
Basechip Logical Devices	**BLD**	Targets physically located in the MEC1609/MEC1609i. The Keyboard Controller Interface uses a Port at 60h/64h	DMA & I/O
Basechip Configuration Register	**BCR**	256 byte space per Logical Device accessed by BIOS through the IOCR-Port.	I/O through CR-Port
Companion Logical Devices	**CLD**	Targets physically located in Companion Chips (e.g. FIR, UART, etc.)	I/O &DMA
Companion Configuration Register	**CCR**	Targets physically located in Companion Chips accessed by BIOS.	I/O through CR-Port

6.1.2 BLOCK DIAGRAM

FIGURE 6-1: LPC INTERFACE IN MEC1609/MEC1609I



6.2 Power, Clocks and Resets

6.2.1 POWER DOMAIN

This block is powered by VTR. Although the block is not powered by VCC, the block is also controlled by VCC_PWRGD. When VCC_PWRGD is de-asserted, the LPC bus pins are placed in the same state they assume when VTR is off. LAD[3:0] and SERIRQ are tri-stated, LDRQ# is pulled high, CLKRUN# is unpowered and LFRAME#, LRESET# and LPCPD# are gated high; see Table 6-2, “LPC Bus Pin Behavior on Reset,” on page 117.The LPC block is also placed in a minimal power state.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

6.2.2 CLOCKS

LPC Logical Device Configuration Registers and LPC Logical Device EC-only Registers in this block are clocked by the LPC Bus Clock. The LPC interface itself is clocked by the PCI_CLK clock input.

The clock rate of the LPC Bus Clock is set by the LPC_AHB Clock Divider Register on page 103. In normal operation the LPC_AHB Clock Divider Register should be set to 01h, so that the LPC bus runs at the 64.52MHz MCLK rate.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

6.2.3 RESETS

This block is affected by nSYS_RST, VCC Power Good and LPC RESET.

The assertion of nSYS_RST resets the LPC state machine and all registers to their default values. The AHB Master state machine is also reset to its default value.

VCC Power Good going low resets the LPC state machine. The AHB Master interface that is part of the Host Interface will go to its idle state. Any transaction that is active on the AHB Master when the VCC POR occurs will be terminated in such a way that the AHB subsystem will not be locked up.

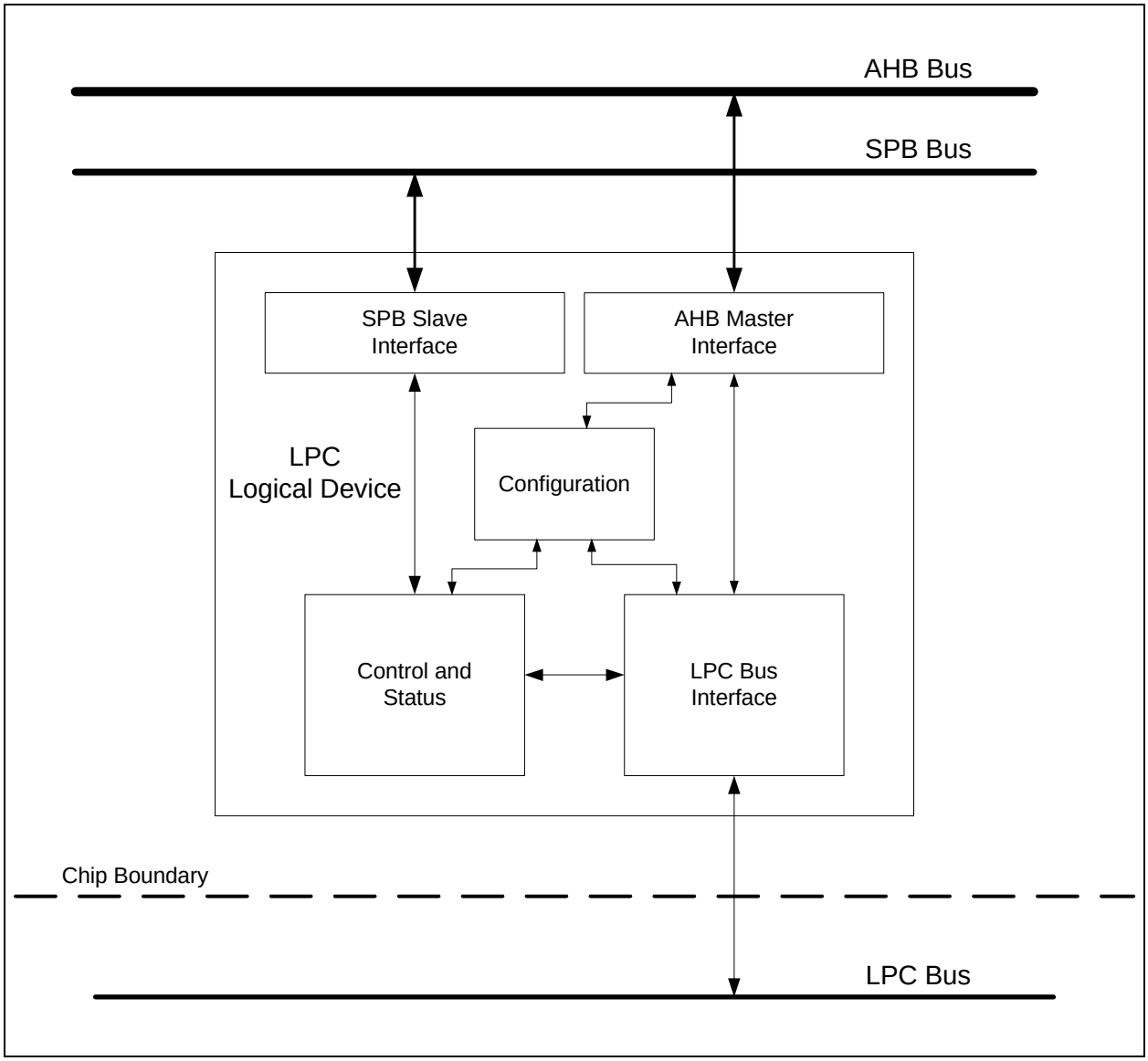
The assertion of LPC RESET# resets the LPC state machine but does not otherwise affect register values. An interrupt to the EC will be generated on either edge of LRESET#. See Section 6.5, "Host Interrupts to EC," on page 120

See Section 5.6, "Reset Interface," on page 95 for details on reset.

6.3 LPC Logical Device

The LPC Logical Device structure is illustrated in Figure 6-1, "LPC Interface in MEC1609/MEC1609i".

FIGURE 6-2: LPC LOGICAL DEVICE



The LPC Logical Device is directly connected to two internal buses, the 32-bit AHB as well as the SPB. The SPB interface is 32-bits. In addition, it is connected to the external LPC bus.

Host accesses to Configuration Registers for each Logical Device on both the MEC1609/MEC1609i and on Companion devices attached to the VLPC bus are managed by a Configuration block described in Section 4.0, "Logical Device Configuration," on page 53. Configuration registers are accessed through the LPC IO Configuration Register Port. The LPC Logical Device translates the Configuration address to an AHB address and the Host LPC access is converted into an AHB transaction inside the MEC1609/MEC1609i.

Host I/O accesses to Configuring Runtime Register Addresses as well as LPC DMA accesses are converted directly to AHB accesses. The LPC address is translated by the LPC Bus Interface to an AHB address inside the MEC1609/MEC1609i and the access becomes an access on the AHB bus.

6.3.1 LPC BUS INTERFACE

The MEC1609/MEC1609i communicates with the host over a Low Pin Count (LPC) interface. The LPC interface uses 3.3V signaling. For detailed specifications, see the *Intel Low Pin Count Specification* and the P*CI Local Bus Specification*, Section 4.2.2. The LPC Bus Interface is listed in Table 2-7, “Host Interface,” on page 13.

The following cycle types are supported by the LPC Bus protocol.

- 8-bit I/O Read
- 8-bit I/O Write
- 8-bit DMA Read (for Logical Devices which support 8-bit DMA)
- 8-bit DMA Write (for Logical Devices which support 8-bit DMA)
- 16-bit DMA Read (for Logical Devices which support 16-bit DMA)
- 16-bit DMA Write (for Logical Devices which support 16-bit DMA)
- 8-bit Trusted I/O Read
- 8-bit Trusted I/O Write

LPC transactions that access registers located on the basechip will require a minimum of two wait SYNCs on the LPC bus. The number of SYNCs may be larger if the AHB bus is in use by the embedded controller, of if the data referenced by the host is not present in a MEC1609/MEC1609i register. The MEC1609/MEC1609i always uses Long Wait SYNCs, rather than Short Wait SYNCs, when responding to an LPC bus request.

Table 6-2, "LPC Bus Pin Behavior on Reset", shows the behavior of LPC outputs and input/outputs under reset conditions in accordance with the *Intel Low Pin Count Specification* and the P*CI Local Bus*. See Section 2.3.1, "Pin Default State Through Power Transitions," on page 11 for Power transition pin state description and Section 6.7, "LPC Clock Run and LPC Power Down Behavior," on page 120 for LPC protocol dependent pin state transitions requirements.

TABLE 6-2: LPC BUS PIN BEHAVIOR ON RESET

Pins	VTR POR (nSYS_RST)	VCC POR	LPCPD# Asserted	LRESET# Asserted
LAD[3:0]	Tri-state	Tri-state	Tri-State	Tri-State
LDRQ#	Tri-state	De-asserted (high)	Tri-State	De-asserted (high)
SERIRQ	Tri-state	Tri-state	Tri-State	Tri-State
CLKRUN#	Tri-state	Tri-state	Tri-State	Tri-State

6.3.2 LPC I/O CYCLES

LPC 8-bit I/O Read cycles and 8-bit I/O Write cycles are mapped directly to addresses in the MEC1609/MEC1609i AHB address space. The mapping will be to either the range FF_0000h through FF_FFFFh, for register addresses located on the MEC1609/MEC1609i basechip, or to the range FE_0000h through FE_FFFFh, for register addresses located on a Companion chip attached to the VLPC Bus. For information on how addresses map between the LPC bus and the MEC1609/MEC1609i, see Section 3.0, "Bus Hierarchy," on page 44 and Section 4.0, "Logical Device Configuration," on page 53. For a list of all Configuration Registers accessible to the Host, see Section 4.0, "Logical Device Configuration," on page 53**.**

6.3.3 LPC FIRMWARE HUB AND MEMORY CYCLES

The MEC1609/MEC1609i does not support LPC Firmware Hub cycles and LPC Memory cycles on the LPC Bus.

6.3.4 LPC DMA CYCLES

LPC DMA Cycles are translated by the MEC1609/MEC1609i into AHB I/O, with DMA Reads translated into AHB writes and DMA Writes translated into AHB reads. The target address of a translated DMA transaction will typically be the address of a FIFO that manages the data The MEC1609/MEC1609i supports DMA transactions to peripherals in Companion devices on the VLPC bus.

6.3.4.1 DMA Channel Access

The LPC Logical Device in the MEC1609/MEC1609i contains a set of registers that correspond to each LPC DMA channel, including channel 4, which is never used (see Section 4.7.1, "DMA Configuration Registers," on page 60). The configuration register for a channel maps the channel number to an AHB address, which should be the address of a FIFO. The configuration mechanism allows any logical device, both on the basechip and in a Companion on the VLPC bus, to be the target of a DMA access.

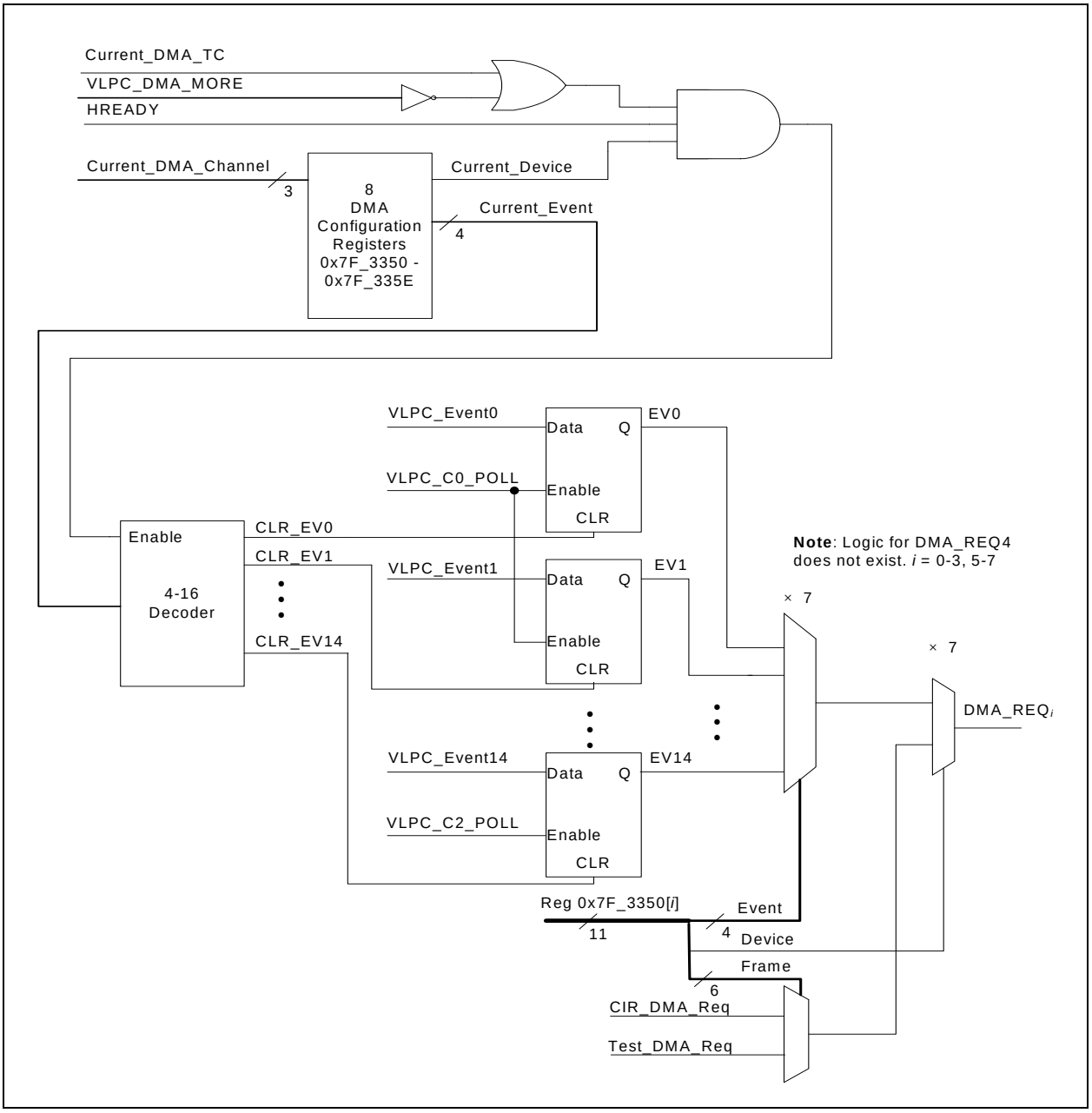
6.3.4.2 DMA Access to Undefined Addresses

If a DMA channel is configured with an undefined or reserved AHB address, LPC DMA transactions on that channel will be terminated with a SYNC cycle indicating no further DMA transfers are required. If the transfer is a DMA Write (from the MEC1609/MEC1609i to the Host), the transfer will return all 1's on the data bus. If the transfer is a DMA Read (from the Host to the MEC1609/MEC1609i), the transfer terminates without any state modification.

6.3.4.3 DMA Request

The LPC Logical Device has seven internal DMA request signals (DMA_REQ_i) that are generated from either basechip DMA peripherals or from the Event signals that are passed between Companion devices on the VLPC bus and the basechip. The generation of these signals is shown in Figure 6-3, "DMA Request Generation". DMA requests from Companion devices are derived from the Event signals that are periodically polled by the basechip.

FIGURE 6-3: DMA REQUEST GENERATION



Note: This figure is a representation of the DMA request circuitry and is not intended to represent actual implementation.

6.3.4.4 DMA Abandonment

When a DMA transaction is abandoned, the DMA Interface is disabled. See Intel Low Pin Count Specification, Revision 1.0, September 29, 1997, Section 6.3 for more information regarding abandoning DMA requests.

DMA Abandonment occurs when the peripheral desires to abandon a DMA transfer. Generally, the cause of DMA Abandonment is peripheral device FIFO overrun or underrun or software stopping a device prematurely. In these cases, the peripheral wishes to stop further DMA activity by sending an LDRQ# message with the ACT bit as '0'. However, since

the DMA request was seen by the host, there is no guarantee that the cycle hasn't been granted and will shortly run on LPC. Therefore, The MEC1609/MEC1609i LPC Bus Interface must take into account that a DMA cycle may still occur. The MEC1609/MEC1609i LPC Bus Interface will choose to complete the cycle normally with any random data. LPC DMA reads (data from host to peripheral) are completed without forwarding. LPC DMA writes (data from peripheral to host) are completed with data field equal 00h.

6.3.5 VLPC BUS FORWARDING

The Very Low Pin Count Bus (VLPC Bus) is a proprietary interconnect designed to enable communication between a master device and up to three peripheral slaves called Companions.

LPC Trusted cycle reads and writes, I/O read and writes, and 8 bit and 16 bit DMA reads and writes can all be forwarded onto the VLPC bus.

All LPC cycles with targets physically located in a Companion chip are mapped into the VLPC Bus address space include some but not all cycles targeting the IOCR-Port. The LPC AHB bus is used to forward transactions from the LPC bus to the VLPC bus.

See Section 8.0, "VLPC Bus Interface," on page 153 methods of controlling the VLPC Bus and Section 4.0, "Logical Device Configuration," on page 53. for a detailed description of the MEC1609/MEC1609i configuration space and registers.

6.3.6 WAIT SYNC'S ON LPC

LPC cycles, with targets physically located in the MEC1609/MEC1609i basechip, are completed with no more than **two** LPC Long WAIT SYNC's. If the EC is accessing a Logical Device located in a Companion chip on the VLPC bus when the Host attempts to access the VLPC bus, the number of Wait SYNC's could be greater.

LPC cycles, with targets physically located in a companion chip, are forwarded to the VLPC-Port and completed on the VLPC before being completed on the LPC. While waiting for transactions to complete on the VLPC, the LPC will have at least extended number of Long WAIT SYNC's.

6.4 LPC Bus Configuration

The mapping from LPC Bus cycles to AHB read/write cycles is managed by the LPC Logical Device. The mapping is defined by a series of configuration registers which are defined in Section 4.0, "Logical Device Configuration," on page 53, in Section 6.4, "LPC Bus Configuration," on page 120.

6.5 Host Interrupts to EC

The LRESET# reset signal and the LPCPD# power down signal can be used to generate EC interrupts and wake-up events. The edge detection of the interrupt and wake events are controlled by their associated Pin Control Register on page 337. The interrupts are routed to the LRESET# and LPCPD# bits in the GIRQ14 Source Register on page 274.

The LPC Logical Device can generate an additional interrupt to the EC when a Host access is mapped to the AHB bus. Bit LPC_AHB_ERR in the Host Bus Error Register is set when an LPC-sourced AHB bus access causes an error; it is also routed to the LPC_AHB_ERR bit in the GIRQ14 Source Register on page 274. For details see Section 6.10.2, "Host Bus Error Register," on page 128.

6.6 EC Interrupts to Host

The Embedded Controller can send an interrupt to the Host on any Serial Interrupt Request channel using the EC SERIRQ Register in conjunction with the SERIRQ Configuration Registers.

6.7 LPC Clock Run and LPC Power Down Behavior

The LPCPD# signal (see the *Intel Low Pin Count Specification*, Section 8.1) and the CLKRUN# signal (see the Intel Low Pin Count Specification, Section 8.2) are implemented in the MEC1609/MEC1609i.

6.7.1 USING LPCPD#

The MEC1609/MEC1609i tolerates the LPCPD# signal going active and then inactive again without LRESET# going active. This is a requirement for notebook power management functions.

The *LPC Bus Specification, Rev. 1.0*, Section 8.2 states that “After LPCPD# goes back inactive, the LPC I/F will always be reset using LRST#”. This text must be qualified for mobile systems where it is possible that when exiting a “light” sleep state (ACPI S1, APM POS), LPCPD# may be asserted but the LPC Bus power may not be removed, in which case LRESET# will not occur. When exiting a “deeper” sleep state (ACPI S3-S5, APM STR, STD, soft-off), LRESET# will occur.

The LPCPD# pin is implemented as a “local” powergood for the LPC bus in the MEC1609/MEC1609i. It is not to be used as a global powergood for the chip. It is used to minimize the LPC power dissipation.

Prior to going to a low-power state, the system asserts the LPCPD# signal. LPCPD# goes active at least 30 microseconds prior to the LCLK signal stopping low and power being shut to the other LPC interface signals. Upon recognizing LPCPD# active, there are no further transactions on the LPC interface. The MEC1609/MEC1609i drives the LDRQ# signal tri-state, and does so until LPCPD# goes active. This prevents the MEC1609/MEC1609i from driving the signals high into a potentially powered-down host.

Upon recognizing LPCPD# inactive, The MEC1609/MEC1609i drives LDRQ# high.

6.7.2 USING CLKRUN#

CLKRUN# is used to indicate the PCI clock status as well as to request that a stopped clock be started. See FIGURE 6-4: CLKRUN# System Implementation Example on page 122, an example of a typical system implementation using CLKRUN#.

The CLKRUN# signal in the MEC1609/MEC1609i also supports the LPC LDRQ# DMA protocol since PCI clock is also required to drive the LDRQ# signal active (See Section 4.7.1, "DMA Configuration Registers," on page 60). If an interrupt or DMA occurs while the PCI clock is stopped, CLKRUN# must be asserted before the interrupt or DMA can be serviced.

PCI Clock Run Support can be enabled and disabled using the Bit[2] SerIRQ Mode in the Device Mode register, Global Configuration Register 24h (see Table 4-19, “Chip-Level (Global) Control/Configuration Registers,” on page 71). When the Bit[2] SerIRQ Mode is ‘0,’ Serial IRQs are disabled, the CLKRUN# pin is disabled, and the affects of Interrupt and DMA requests on CLKRUN# are ignored. When the Bit[2] SerIRQ Mode is ‘1,’ Serial IRQs are enabled, the CLKRUN# pin is enabled, and the CLKRUN# support related to Interrupts and DMA requests as described in the section below is enabled.

The CLKRUN# pin is an open drain output and input. Refer to the *PCI Mobile Design Guide Rev 1.0* for a description of the CLKRUN# function. If CLKRUN# is sampled “high”, the PCI clock is stopped or stopping. If CLKRUN# is sampled “low”, the PCI clock is starting or started (running). CLKRUN# in the MEC1609/MEC1609i supports both Serial IRQ and LPC DMA cycles.

6.7.2.1 CLKRUN# Support for Serial IRQ Cycle

If a device in the MEC1609/MEC1609i asserts or de-asserts an interrupt and CLKRUN# is sampled “high”, the MEC1609/MEC1609i can request the restoration of the clock by asserting the CLKRUN# signal asynchronously (Table 6-3). The MEC1609/MEC1609i holds CLKRUN# low until it detects two rising edges of the clock. After the second clock edge, the MEC1609/MEC1609i must disable the open drain driver (Figure 6-5).

The MEC1609/MEC1609i must not assert CLKRUN# if it is already driven low by the central resource; i.e., the PCI CLOCK GENERATOR in Figure 6-5. The MEC1609/MEC1609i will not assert CLKRUN# under any conditions if the Serial IRQs are disabled.

The MEC1609/MEC1609i must not assert CLKRUN# unless the line has been de-asserted for two successive clocks; i.e., before the clock was stopped (Figure 6-5).

6.7.2.2 CLKRUN# Support for LPC DMA Cycle

If a device in the MEC1609/MEC1609i requests DMA service while the PCI clock is stopped, CLKRUN# must be asserted to restart the PCI clock so that the LDRQ# signal may be asserted (See Table 6-3). The MEC1609/MEC1609i will not assert CLKRUN# under any conditions if the SerIRQ_Mode bit is inactive (“0”).

If a device in the MEC1609/MEC1609i asserts a DMA request and CLKRUN# is sampled “high”, the MEC1609/MEC1609i holds CLKRUN# low until it detects two rising edges of the PCI clock. After the second clock edge, the MEC1609/MEC1609i must disable the CLKRUN# open-drain driver (See Figure 6-5).

The MEC1609/MEC1609i will not assert CLKRUN# if it is already driven low by the central resource; i.e., the PCI CLOCK GENERATOR. The MEC1609/MEC1609i also will not assert CLKRUN# unless the signal has been de-asserted for two successive clocks; i.e., before the clock was stopped.

TABLE 6-3: MEC1609/MEC1609I CLKRUN# FUNCTION

SIRQ_MODE (Bit[2] SerIRQ Mode in Device Mode Register)	Internal Interrupt or DMA Request	CLKRUN#	Action
0	X	X	None
1	NO CHANGE	X	None
	CHANGE (Note 6-1)	0	None
		1	Assert CLKRUN#

Note 6-1 "Change" means either-edge change on any or all parallel IRQs routed to the Serial IRQ block. "Assertion" means assertion of DMA request by a device in the MEC1609/MEC1609i. The "change" detection logic must run asynchronously to the PCI Clock and regardless of the Serial IRQ mode; i.e., "continuous" or "quiet".

FIGURE 6-4: CLKRUN# SYSTEM IMPLEMENTATION EXAMPLE

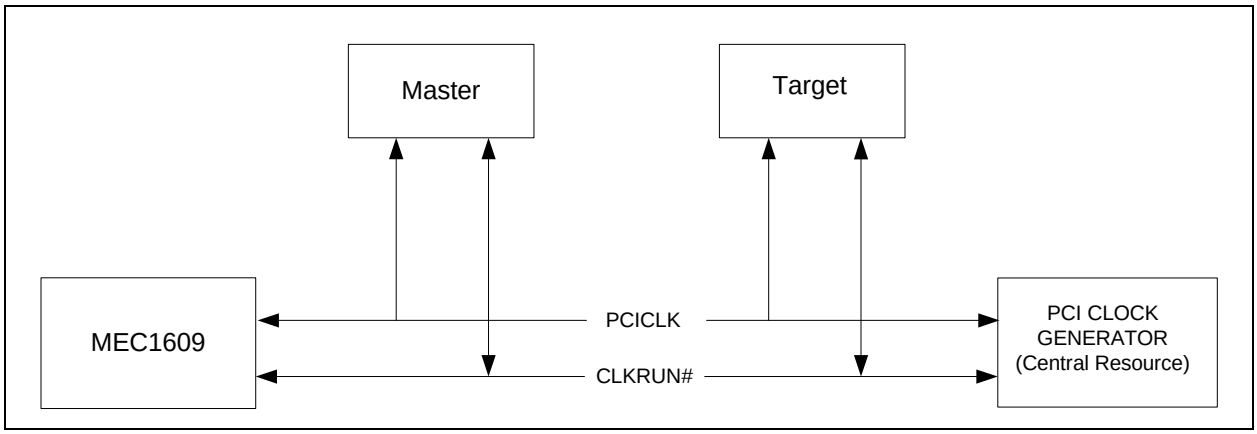
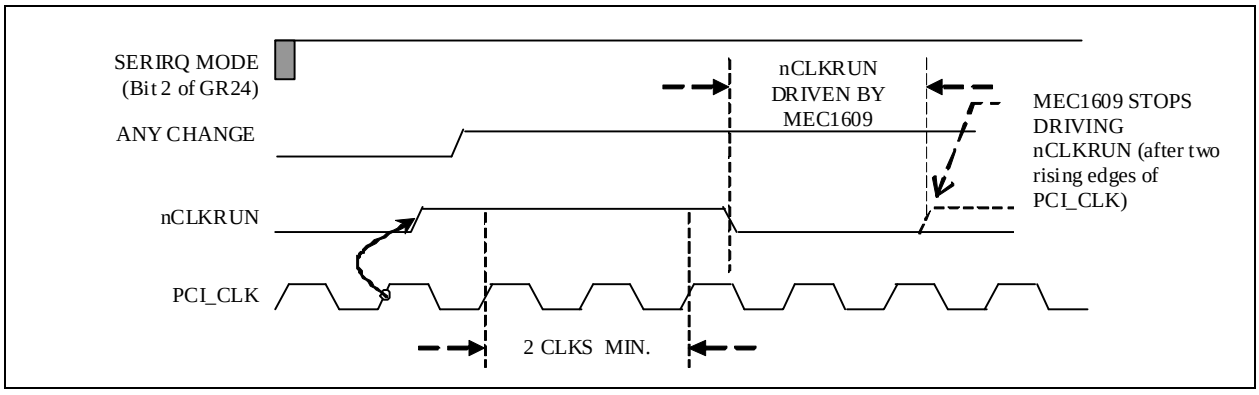


FIGURE 6-5: CLOCK START ILLUSTRATION



Note:

- The signal “ANY CHANGE” is the same as “CHANGE/ASSERTION” in Table 6-3.
- The MEC1609/MEC1609i must continually monitor the state of CLKRUN# to maintain the PCI Clock until an active “any IRQ change” condition has been transferred to the host in a Serial IRQ cycle or “any DRQ assertion” condition has been transferred to the host in a DMA cycle. For example, if “any IRQ change or DRQ assertion” is asserted before CLKRUN# is de-asserted (not shown in Figure 6-5), the MEC1609/MEC1609i must assert CLKRUN# as needed until the Serial IRQ cycle or DMA cycle has completed.

6.8 Using Serial Interrupts

The MEC1609/MEC1609i will support the serial interrupt scheme, which is adopted by several companies, to transmit interrupt information to the system. The serial interrupt scheme adheres to the *Serial IRQ Specification for PCI Systems Version 6.0*.

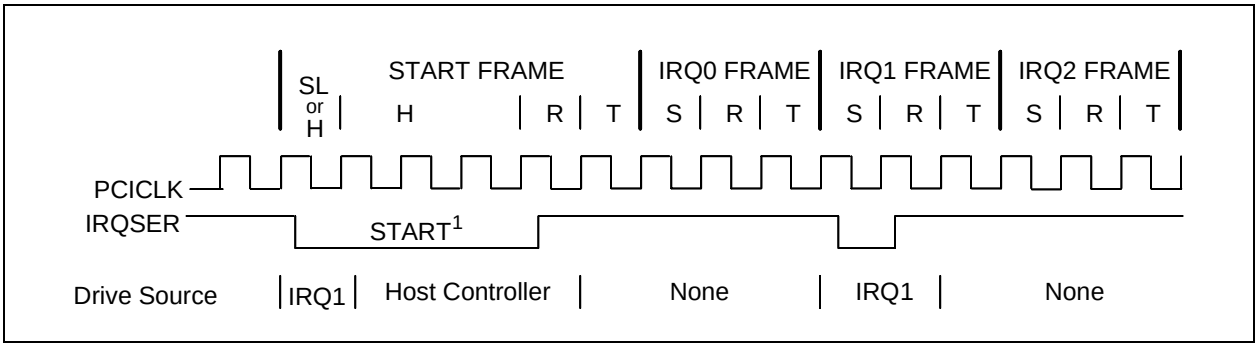
TIMING DIAGRAMS for IRQSER CYCLE

PCICLK = 33 MHz_IN pin

IRQSER = SIRQ pin

Start Frame timing with source sampled a low pulse on IRQ1.

FIGURE 6-6: SERIAL INTERRUPTS WAVEFORM “START FRAME”

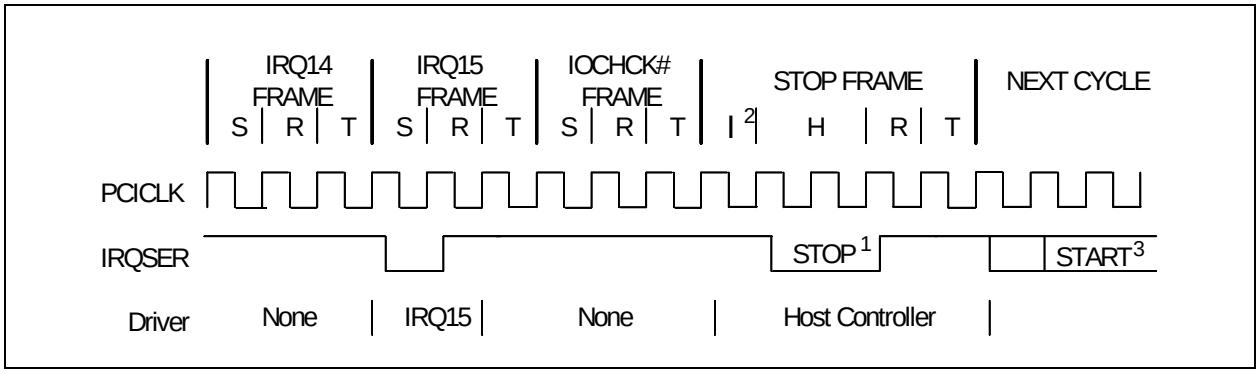


H=Host Control SL=Slave Control R=Recovery T=Turn-around S=Sample

Start Frame pulse can be 4-8 clocks wide.

Stop Frame Timing with Host using 17 IRQSER sampling period.

FIGURE 6-7: SERIAL INTERRUPT WAVEFORM "STOP FRAME"



H=Host Control R=Recovery T=Turn-around S=Sample I= Idle

Stop pulse is two clocks wide for Quiet mode, three clocks wide for Continuous mode.

There may be none, one, or more Idle states during the Stop Frame.

The next IRQSER cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

6.8.1 SERIRQ MODE BIT FUNCTION

TABLE 6-4: SERIRQ_EN CONFIGURATION CONTROL

CR25 BIT[2]	Name	Description
0	SERIRQ_EN	Serial IRQ Disabled
1		Serial IRQ Enabled (Default)

6.8.1.1 IRQSER Cycle Control

There are two modes of operation for the IRQSER Start Frame.

Quiet (Active) Mode

Any device may initiate a Start Frame by driving the IRQSER low for one clock, while the IRQSER is Idle. After driving low for one clock, the IRQSER must immediately be tri-stated without at any time driving high. A Start Frame may not be initiated while the IRQSER is active. The IRQSER is Idle between Stop and Start Frames. The IRQSER is active between Start and Stop Frames. This mode of operation allows the IRQSER to be Idle when there are no IRQ/Data transitions which should be most of the time.

Once a Start Frame has been initiated, the host controller will take over driving the IRQSER low in the next clock and will continue driving the IRQSER low for a programmable period of three to seven clocks. This makes a total low pulse width of four to eight clocks. Finally, the host controller will drive the IRQSER back high for one clock then tri-state.

Any IRQSER Device (i.e., The MEC1609/MEC1609i) which detects any transition on an IRQ/Data line for which it is responsible must initiate a Start Frame in order to update the host controller unless the IRQSER is already in an IRQSER Cycle and the IRQ/Data transition can be delivered in that IRQSER Cycle.

Continuous (Idle) Mode

Only the Host controller can initiate a Start Frame to update IRQ/Data line information. All other IRQSER agents become passive and may not initiate a Start Frame. IRQSER will be driven low for four to eight clocks by host controller. This mode has two functions. It can be used to stop or idle the IRQSER or the host controller can operate IRQSER in a continuous mode by initiating a Start Frame at the end of every Stop Frame.

An IRQSER mode transition can only occur during the Stop Frame. Upon reset, IRQSER bus is defaulted to continuous mode, therefore only the host controller can initiate the first Start Frame. Slaves must continuously sample the Stop Frames pulse width to determine the next IRQSER Cycle's mode.

IRQSER Data Frame

Once a Start Frame has been initiated, the MEC1609/MEC1609i will watch for the rising edge of the Start Pulse and start counting IRQ/Data Frames from there. Each IRQ/Data Frame is three clocks: Sample phase, Recovery phase, and Turn-around phase. During the sample phase, the MEC1609/MEC1609i must drive the IRQSER (SIRQ pin) low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, IRQSER must be left tri-stated. During the recovery phase, the MEC1609/MEC1609i must drive the SERIRQ high, if and only if, it had driven the IRQSER low during the previous sample phase. During the turn-around phase, the MEC1609/MEC1609i must tri-state the SERIRQ. The MEC1609/MEC1609i drives the IRQSER line low at the appropriate sample point if its associated IRQ/Data line is low, regardless of which device initiated the start frame.

The Sample phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one e.g. The IRQ5 Sample clock is the sixth IRQ/Data Frame, then the sample phase is {(6 x 3) - 1 = 17} the seventeenth clock after the rising edge of the Start Pulse.

TABLE 6-5: IRQSER SAMPLING PERIODS

IRQSER Period	Signal Sampled	# of Clocks Past Start
1	Not Used	2
2	IRQ1	5
3	nSMI/IRQ2	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47

The SIRQ data frame will now support IRQ2 from a logical device; previously IRQSER Period 3 was reserved for use by the System Management Interrupt (nSMI). When using Period 3 for IRQ2, the user should mask off the MEC1609/MEC1609i's SMI via the ESMI Mask Register. Likewise, when using Period 3 for nSMI, the user should not configure any logical devices as using IRQ2.

IRQSER Period 14 is used to transfer IRQ13. Each Logical devices will have IRQ13 as a choice for their primary interrupt.

Stop Cycle Control

Once all IRQ/Data Frames have completed, the host controller will terminate IRQSER activity by initiating a Stop Frame. Only the host controller can initiate the Stop Frame. A Stop Frame is indicated when the IRQSER is low for two or three clocks. If the Stop Frame's low time is two clocks, then the next IRQSER cycle's sampled mode is the Quiet mode; and any IRQSER device may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse. If the Stop Frame's low time is three clocks, then the next IRQSER cycle's sampled mode is the continuous mode, and only the host controller may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse.

Latency

Latency for IRQ/Data updates over the IRQSER bus in bridge-less systems with the minimum IRQ/Data Frames of 17 will range up to 96 clocks (3.84µS with a 25 MHz PCI Bus or 2.88µs with a 33 MHz PCI Bus). If one or more PCI to PCI Bridge is added to a system, the latency for IRQ/Data updates from the secondary or tertiary buses will be a few clocks longer for synchronous buses, and approximately double for asynchronous buses.

EOI/ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR Read to precede an IRQ transition that it should have followed. This could cause a system fault. The host interrupt controller is responsible for ensuring that these latency issues are mitigated. The recommended solution is to delay EOIs and ISR Reads to the interrupt controller by the same amount as the IRQSER Cycle latency in order to ensure that these events do not occur out of order.

AC/DC Specification Issue

All IRQSER agents must drive/sample IRQSER synchronously related to the rising edge of the PCI bus clock. The IRQSER (SIRQ) pin uses the electrical specification of the PCI bus. Electrical parameters will follow the PCI Specification Section 4, sustained tri-state.

Reset and Initialization

The IRQSER bus uses LRESET as its reset signal and follows the PCI bus reset mechanism. The IRQSER pin is tri-stated by all agents while LRESET is active. With reset, IRQSER slaves and bridges are put into the (continuous) Idle mode. The host controller is responsible for starting the initial IRQSER cycle to collect system's IRQ/Data default values. The system then follows with the Continuous/Quiet mode protocol (Stop Frame pulse width) for subsequent IRQSER cycles. It is the host controller's responsibility to provide the default values to the 8259's and other system logic before the first IRQSER cycle is performed. For IRQSER system suspend, insertion, or removal application, the host controller should be programmed into Continuous (IDLE) mode first. This is to guarantee the IRQSER bus is in Idle state before the system configuration changes.

6.9 LPC Logical Device Configuration Registers

The configuration registers in the LPC Logical Device are described in Section 4.0, "Logical Device Configuration," on page 53. These registers control the activity of all the Logical Devices in the MEC1609/MEC1609i. The Activate Register controls the LPC device itself. The Host can shut down the LPC Logical Device by clearing the Activate bit, but it cannot restart the LPC interface, since once the LPC interface is inactive the Host has no access to any registers on the MEC1609/MEC1609i. The Embedded Controller can set or clear the Activate bit at any time.

TABLE 6-6: ACTIVATE REGISTER

HOST OFFSET	BYTE0: 30h			8-bit			**HOST SIZE**	
EC ADDRESS	FF_3330h			8-bit			**EC SIZE**	
POWER	VTR			00b			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the LPC Logical Device is powered and functional.

When this bit is 0, the logical device is powered down and inactive. Except for the Activate Register itself, clocks to the block are gated and the LPC Logical Device will permit the ring oscillator to be shut down (see Section 6.10.4, "EC Clock Control Register," on page 130). LPC bus output pads will be tri-stated. Serial IRQ activation is separately controlled by the Device Mode register in the Chip-Level (Global) Control/Configuration Registers.

APPLICATION NOTE: The Activate bit in the Activate Register should not be written '0' to by the Host over LPC.

6.10 LPC Logical Device EC-only Registers

Table 6-7, "LPC EC-only Registers" summarizes the registers in the Host Interface block that are only accessible by the EC. In addition to these registers, the Host Interface also contains Configuration registers, described in Section 4.0, "Logical Device Configuration," on page 53.

TABLE 6-7: LPC EC-ONLY REGISTERS

AHB Address	Name	VTR POR (nSYS_RST) Default
FF_3100h	Reserved	0000_0000h
FF_3104h	LPC Bus Monitor Register	0000_0000h
FF_3108h	Host Bus Error Register	0000_0000h
FF_310Ch	EC SERIRQ Register	0000_0000h
FF_3110h	EC Clock Control Register	0000_0000h
FF_3120h	BAR Inhibit Register	0000_0000h
FF_3128h	External BAR Inhibit Device Map	0000_0000h

Because their addresses are in the part of the LPC Logical Device address frame that is not addressable from the LPC bus, the following registers are accessible only to the EC.

6.10.1 LPC BUS MONITOR REGISTER

TABLE 6-8: LPC BUS MONITOR REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC ADDRESS	FF_3104h				32-bit		**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved						LRESET_ Status	LPCPD_ Status

LPCPD_STATUS

This bit reflects the state of the LPCPD# input pin. The LPCPD_Status is the inverse of the LPCPD# pin (see Section 6.7, "LPC Clock Run and LPC Power Down Behavior," on page 120).

When the LPCPD_Status bit is ‘0b’, the LPCPD# input pin is de-asserted (that is, the pin has the value ‘1b’). When the LPCPD_Status bit is ‘1b’, the LPCPD# input pin is asserted (that is, the pin has the value ‘0b’).

An interrupt to the EC will be generated on either edge of LPCPD#. See Section 6.5, "Host Interrupts to EC," on page 120.

LRESET_STATUS

This bit reflects the state of the LRESET# input pin. The LRESET_Status is the inverse of the LRESET# pin (see Section 6.2.3, "Resets," on page 115).

When the LRESET_Status bit is '0b', the LRESET# input pin is de-asserted (that is, the pin has the value '1b'). When the LRESET_Status bit is '1b', the LRESET# input pin is asserted (that is, the pin has the value '0b').

An interrupt to the EC will be generated on either edge of LRESET#. See Section 6.5, "Host Interrupts to EC," on page 120.

6.10.2 HOST BUS ERROR REGISTER

TABLE 6-9: HOST BUS ERROR REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	FF_3108h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ErrorAddress[23:16]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ErrorAddress[15:8]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ErrorAddress[7:0]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/WC	R/WC	R/WC	R/WC	R/WC	R/W	R/WC
BIT NAME	Reserved	Reserved	DMA Err	Config Err	Runtime Err	BAR_ Conflict	En_ AHB_Err	LPC_ AHB_Err

LPC_AHB_ERR

This bit can be used to generate an EC interrupt. It is set whenever either a BAR conflict or an AHB bus error occurs as a result of an LPC access. Once set, it remains set until cleared by being written with a 1.

EN_AHB_ERR

When this bit is 0, only a BAR conflict, which occurs when two BARs match the same LPC I/O address, will cause LPC_AHB_ERR to be set. When this bit is 1, AHB bus errors will also cause LPC_AHB_ERR to be set.

BAR_CONFLICT

This bit is set to 1 whenever a BAR conflict occurs on an LPC address. A Bar conflict occurs when more than one BAR matches the address during of an LPC cycle access. Once this bit is set, it remains set until cleared by being written with a 1.

RUNTIME_ERR

This bit is set to 1 whenever En_AHB_ERR is 1 and an LPC I/O access causes an AHB bus error. This error will only occur if a BAR is misconfigured. Once set, it remains set until cleared by being written with a 1.

CONFIG_ERR

This bit is set to 1 whenever En_AHB_ERR is 1 and an LPC Configuration access causes an AHB bus error. Once set, it remains set until cleared by being written with a 1.

DMA_ERR

This bit is set to 1 whenever En_AHB_ERR is 1 and an LPC DMA access causes an AHB bus error. Once set, it remains set until cleared by being written with a 1.

ERRORADDRESS

This 24-bit field captures the 24-bit AHB address of every LPC transaction whenever the bit LPC_AHB_ERR in this register is 0. When LPC_AHB_ERR is 1 this register is not updated but retains its previous value. When bus errors occur this field saves the address of the first address that caused an error.

6.10.3 EC SERIRQ REGISTER

TABLE 6-10: EC SERIRQ REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	FF_310Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							EC_IRQ

EC_IRQ

If the LPC Logical Device is selected as the source for a Serial Interrupt Request by an Interrupt Configuration register (see Section 4.8, "SERIRQ Interrupts," on page 62), this bit is used as the interrupt source.

6.10.4 EC CLOCK CONTROL REGISTER

TABLE 6-11: EC CLOCK CONTROL REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	FF_3110h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						Clock_Control	

CLOCK_CONTROL

This field controls when the host interface will permit the internal ring oscillator to be shut down. The choices are as follows:

- **0h:** The host interface will permit the ring oscillator to be shut down if the LPCPD# signal is asserted (sampled low).
- **1h:** The host interface will permit the ring oscillator to be shut down if the CLKRUN# signals “CLOCK STOP” and there are no pending serial interrupt request or DMA requests from devices associated with the MEC1609/MEC1609i. The CLKRUN# signals “CLOCK STOP” by CLKRUN# being high for 5 LPCCLK’s after the raising edge of CLKRUN#.
- **2h:** The host interface will permit the ring oscillator to be shut down after the completion of every LPC transaction. It will require the ring oscillator as soon as a START field appears on the LPC bus. This mode may cause an increase in the time to respond to LPC transactions if the ring oscillator has to turn on and adjust its speed after the LPC transaction arrives at the MEC1609/MEC1609i
- **3h:** The ring oscillator is not permitted to shut down as long as the host interface is active.

When the Activate bit in the Activate Register on page 126 is 0, the Host Interface will permit the ring oscillator to be shut down and the Clock_Control Field is ignored. The Clock_Control Field only effects the Host Interface when the Activate bit in the Activate Register is 1.

6.10.5 BAR INHIBIT REGISTER

TABLE 6-12: BAR INHIBIT REGISTER

HOST OFFSET	n/a			n/a			**HOST SIZE**	
EC OFFSET	120h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		BAR_Inhibit[13:8]					
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	BAR_Inhibit[7:0]							

BAR_INHIBIT

When bit *i* of BAR_Inhibit is 1, the BAR for the associated Logical Device is disabled and its addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR. When bit *i* is 0, BAR activity for Logical Device *i* is based on the Valid bit in the BAR. The association between bits in BAR_Inhibit and Logical Devices is shown in Table 6-13, "BAR Inhibit Device Map".

TABLE 6-13: BAR INHIBIT DEVICE MAP

BAR Inhibit Bit	Logical Device Number	Logical Device	BAR AHB Offset
0	Ch	LPC Interface	360h
1	0h	Mailbox	364h
2	1h	Keyboard Controller	368h
3	2h	ACPI EC Channel 0	36Ch
4	3h	ACPI EC Channel 1	370h
5	4h	ACPI EC Channel 2	374h
6	5h	ACPI EC Channel 3	378h

TABLE 6-13: BAR INHIBIT DEVICE MAP (CONTINUED)

BAR Inhibit Bit	Logical Device Number	Logical Device	BAR AHB Offset
7	6h	ACPI PM1	37Ch
8	7h	UART	380h
9	8h	Legacy	384h
10	Eh	Embedded Flash	388h
11	Fh	Flash SPI	38Ch
12	10h	EM Interface	390h

6.10.6 EXTERNAL BAR INHIBIT REGISTER

TABLE 6-14: EXTERNAL BAR INHIBIT REGISTER

HOST OFFSET	n/a			n/a			**HOST SIZE**	
EC OFFSET	128h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	BAR_Inhibit[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	BAR_Inhibit[7:0]							

BAR_INHIBIT[15:0]

When bit *i* of BAR_Inhibit is 1, the External Logical Device BAR *i* for is disabled and its addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR. When bit *i* is 0, BAR activity for Logical Device *i* is based on the Valid bit in the BAR. The association between bits in BAR_Inhibit and Logical Devices is shown in Table 6-15, "External BAR Inhibit Device Map".

TABLE 6-15: EXTERNAL BAR INHIBIT DEVICE MAP

BAR Inhibit Bit	BAR AHB Offset
0	3B0h
1	3B4h
2	3B8h
3	3BCh
4	3C0h
5	3C4h
6	3C8h
7	3CCh
8	3D0h
9	3D4h
10	3D8h
11	3DCh
12	3E0h
13	3E4h
14	3E8h
15	3ECh

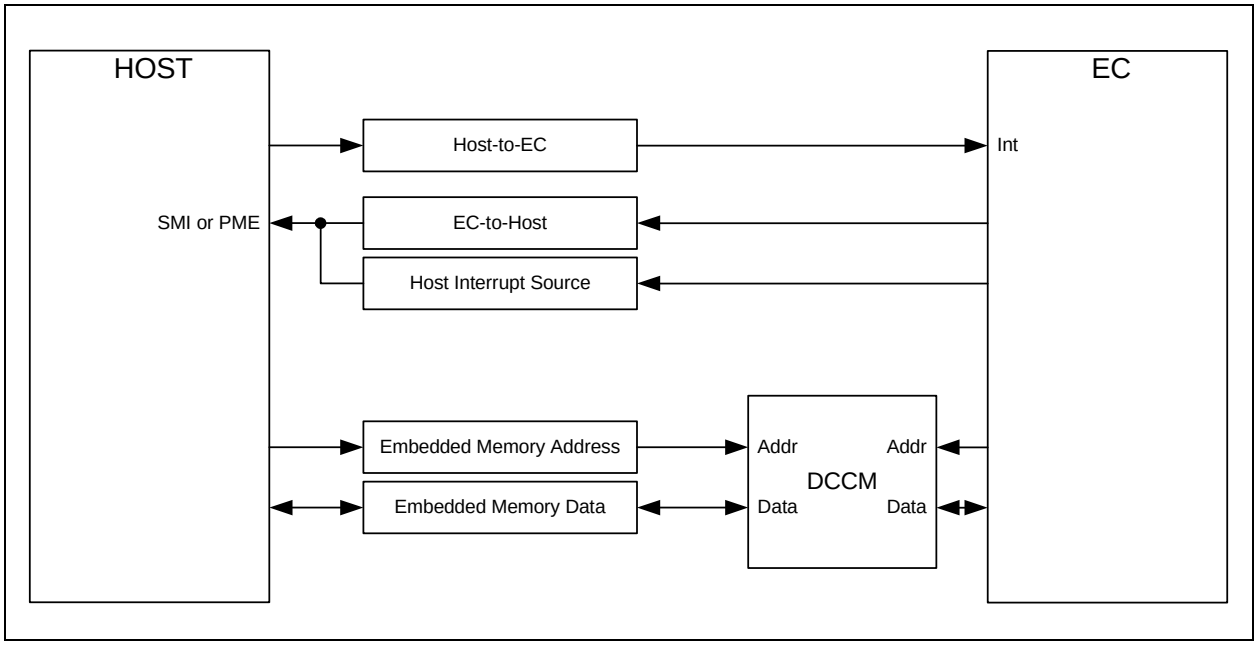
7.0 EMBEDDED MEMORY INTERFACE

7.1 General Description

The Embedded Memory Interface provides a standard run-time mechanism for the host to communicate with the Embedded Controller (EC) and other logical components in the MEC1609/MEC1609i (Figure 7-1). The Embedded Memory Interface includes 12 byte-addressable registers in the Host's I/O address space, as well as 20 bytes of registers that are accessible only by the EC. The Embedded Memory Interface can be used by the Host to read any byte in a region of EC closely-coupled memory, designated by the EC, without requiring any assistance from the EC. A portion of the memory can be written by the Host without any EC assistance as well.

7.1.1 BLOCK DIAGRAM

FIGURE 7-1: EMBEDDED MEMORY INTERFACE BLOCK DIAGRAM



7.2 Power, Clocks and Reset

7.2.1 POWER DOMAIN

This block is powered by the VTR power supply.

7.2.2 CLOCKS

This block has two clock inputs, the LPC Bus Clock and the EC clock.

The Embedded Memory Interface includes support for system-level clock gating. The clock required output is the inversion of the sleep enable input.

7.2.3 RESET

This block is reset when nSYS_RST is asserted.

7.3 Interrupts

Each instance of the Embedded Memory Interface can generate an interrupt event for the HOST-to-EC events. See HOST-to-EC Mailbox Register on page 141. The interrupt source for the EMI is routed onto the EM_MBX bit in the GIRQ15 Source Register and is edge-sensitive, active high.

When the EM_MBX interrupt status bit is cleared while there is data in the HOST-to-EC Mailbox Register, the interrupt remains not asserted until the host writes another byte. If the host writes a second byte to the HOST-to-EC Mailbox Register before the EC processes the EM_MBX interrupt resulting from the first byte, the data from the first byte will be overwritten by the second before the EC has a chance to process the first. This can be avoided with a proper handshaking protocol between the host and EC (for example, the EC should process a data in the HOST-to-EC Mailbox Register before signaling the host that the data register is free).

7.3.1 EMBEDDED MEMORY INTERFACE SIRQ ROUTING

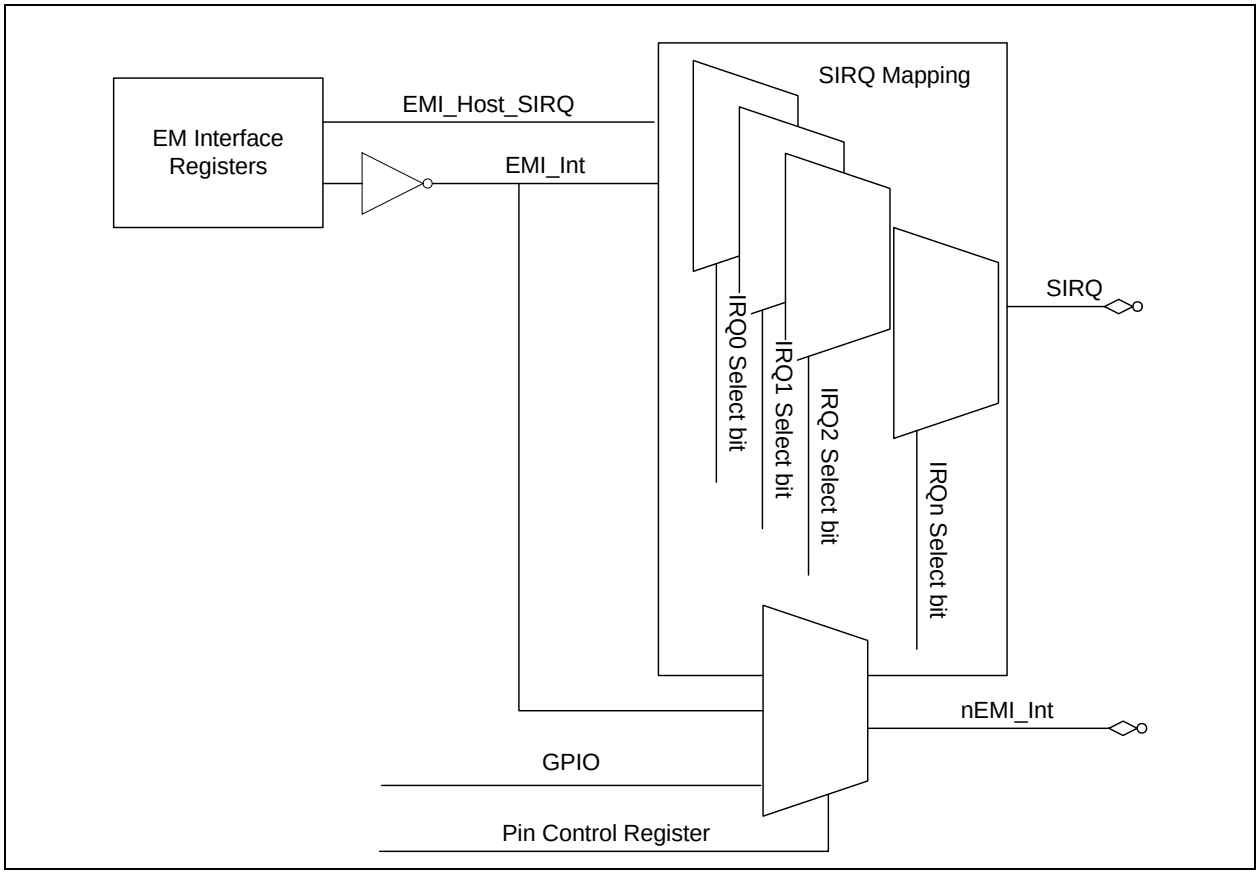
The Embedded Memory Interface can generate a SIRQ event for the EC-to-HOST EC events. See HOST-to-EC Mailbox Register on page 141 and Interrupt Source Register on page 145. This interrupt is routed to the SIRQ block (see SERIRQ Configuration Registers on page 63). For this interrupt, the SELECT on page 64 is cleared to ‘0’ in the Interrupt Configuration Register for the selected SIRQ frame.

The Embedded Memory Interface can also generate an event on an external pin, nEM_Int. The pin can be routed to SMI or PME inputs, as required. The EC can cause the event to be generated by either writing the EC-to-Host Mailbox Register or by setting any of the enabled EC_SWI[14:0] bits in the Interrupt Source Register to 1. The event can be routed to any frame in the SIRQ stream or to the external pin. To enable routing to the SIRQ stream, the bit SELECT on page 64 is set to ‘1’ in the Interrupt Configuration Register for the selected SIRQ frame. The event can be routed to the pin by selecting the EM_Int signal function in the associated Pin Control Register on page 337.

The event produces a standard active low on the serial IRQ stream and active low on the open drain EM_Int pin. See FIGURE 7-2: Embedded Memory Interface SIRQ and nEM_Int routing on page 136.

See Section 4.8.2, "SERIRQ Configuration Registers," on page 63.

FIGURE 7-2: EMBEDDED MEMORY INTERFACE SIRQ AND NEM_INT ROUTING



7.4 Description

The Embedded Memory Interface contains a Mailbox that enables the Host to send an 8-bit message to the EC and the EC to send an 8-bit message to the Host. When written by the sender, the messages can generate an interrupt at the receiver.

In addition to the messages that can be exchanged, the Embedded Memory Interface permits the Host to read and write a portion of the EC's Data Closely Coupled Memory (DCCM). Host reads and writes take place without intervention or assistance from the EC.

The Embedded Memory Interface occupies 12 bytes in the Host I/O space. Two bytes constitute the Host-to-EC and EC-to-Host message links. Six bytes are used for the interface into the EC DCCM, two for address and four for data. The four data bytes are used for reads and writes to the EC DCCM DMI.

When the Host reads one of the four bytes in the Embedded Memory Interface data register, data from the DCCM at the address defined by the Embedded Memory Interface address register is returned to the Host. Writes to a byte write the corresponding byte in the DCCM. The Embedded Memory Interface can be configured so that, although Host I/O is always byte at a time, transfers between the Embedded Memory Interface data bytes and the DCCM can configured to occur as single bytes, 2-byte blocks or 4-byte blocks. This is done so that data that the EC treats as 16-bit or 32-bit will be consistent in the Host, even though one byte of the DCCM data may change between two or more 8-bit accesses by the Host.

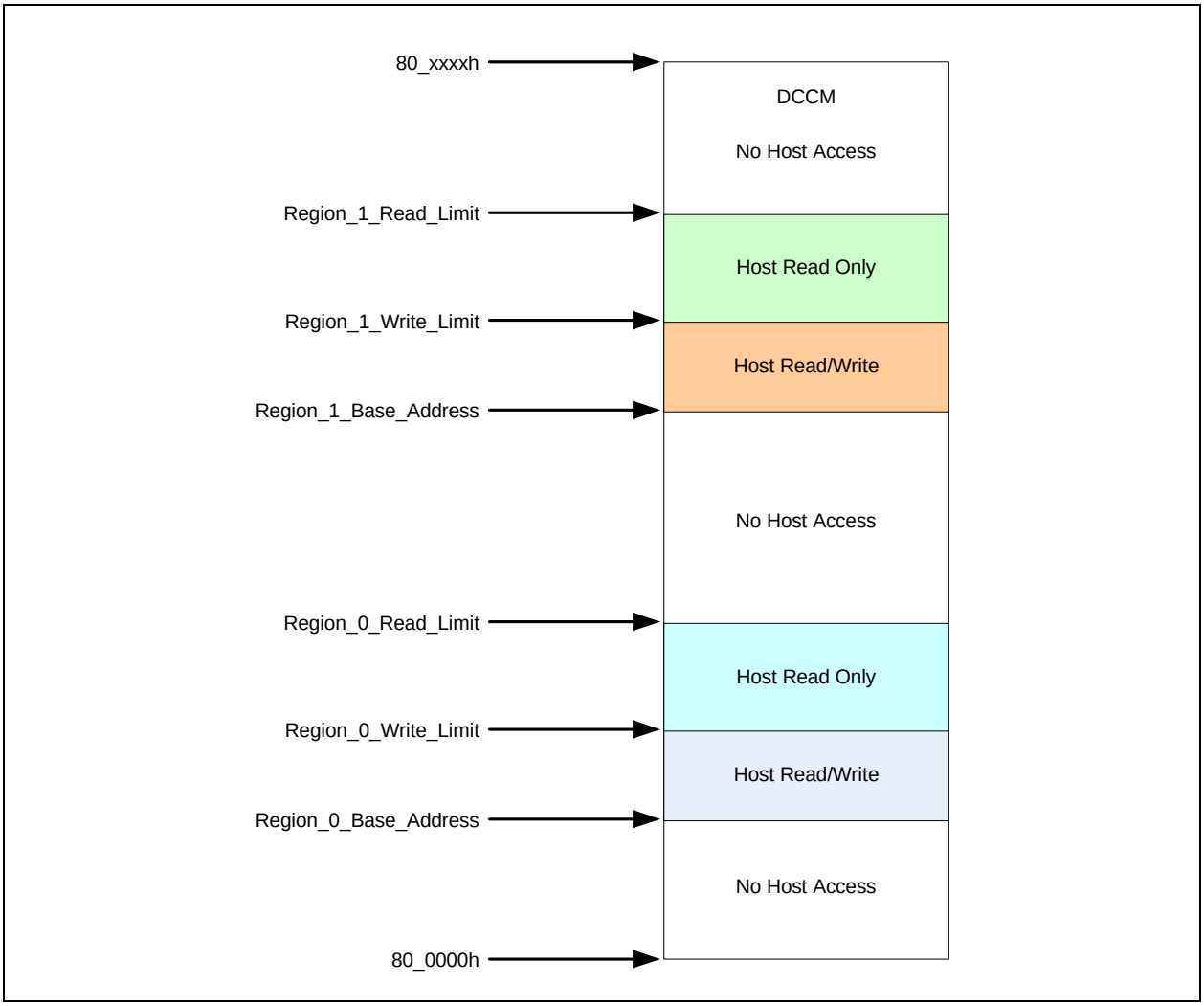
In addition, there is an auto-increment function for the Embedded Memory Interface address register. When enabled, the Host can read or write blocks of memory in the DCCM by repeatedly accessing the Embedded Memory Interface data register, without requiring Host updates to the Embedded Memory Interface address register.

APPLICATION NOTE: the RAM_Select bit in the AHB SRAM Configuration Register must be asserted ('1') to properly configure the DCCM for use by the Embedded Memory Interface.

7.4.1 EMBEDDED MEMORY MAP

Each Embedded Memory interface provides direct access for the Host into two windows in the EC DCCM SRAM. This mapping is shown in Figure 7-3, "Embedded Memory Addressing":

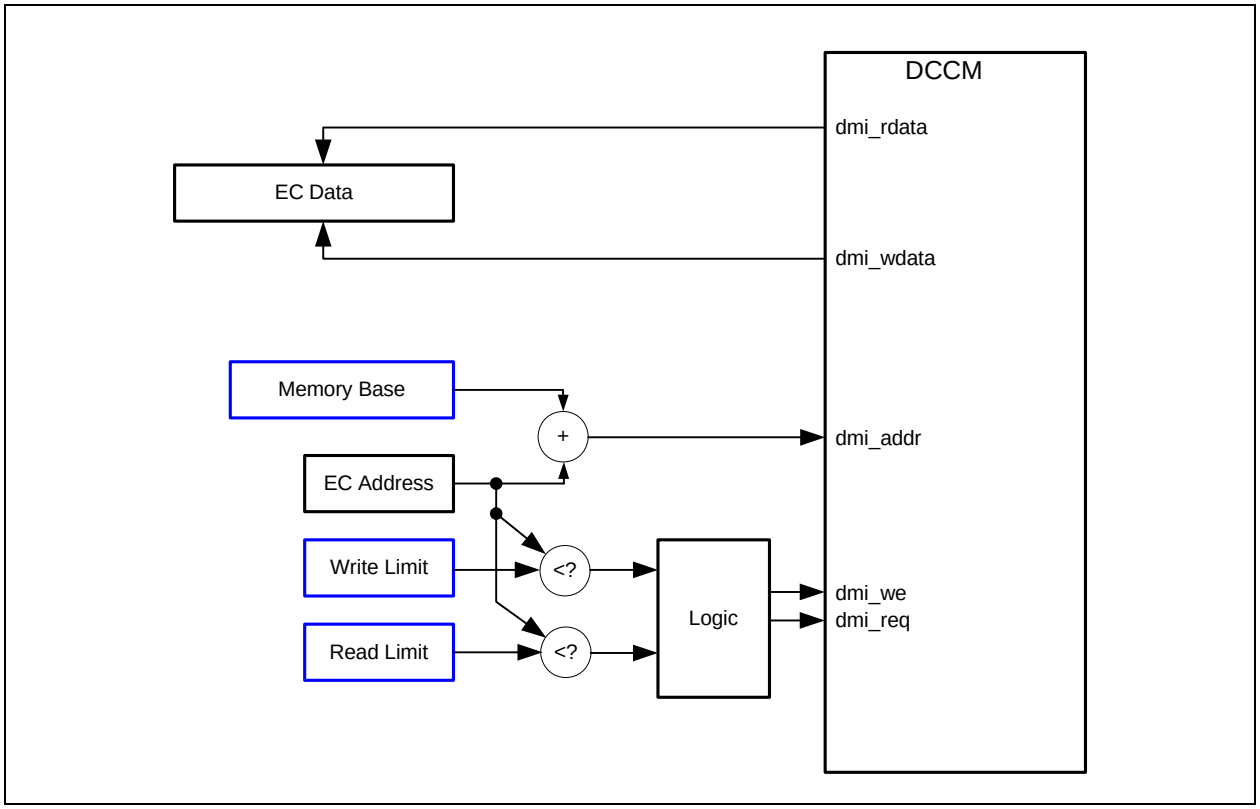
FIGURE 7-3: EMBEDDED MEMORY ADDRESSING



The Base addresses, the Read limits and the Write limits are defined by registers that are in the EC address space and cannot be accessed by the Host. In each region, the Read limit need not be greater than the Write limit. The regions can be contiguous or overlapping.For example, if the Region 0 Read limit is set to 0 and the Write limit is set to a positive number, then the Embedded Memory interface defines a region in the EC memory that the EC can read and write but is write-only for the host. This might be useful for storage of security data, which the Host might wish to send to the EC but should not be readable in the event a virus invades the Host.

Each window into the EC memory can be as large as 32K bytes. The Embedded Memory Interface uses the EC's DCCM Direct Memory Interface (DMI) in order to access the memory. Figure 7-4, "Embedded Memory Region Address Control" shows the relationship between one of the regions in the Embedded Memory Interface and the DCCM DMI:

FIGURE 7-4: EMBEDDED MEMORY REGION ADDRESS CONTROL



7.4.2 EMBEDDED MEMORY INTERFACE USAGE

The Embedded Memory Interface provides a generic facility for communication between the Host and the EC and can be used for many functions. Some examples are:

- Virtual registers. A block of read-only memory locations in the DCCM can be used to implement a set of virtual registers. The EC can update these locations with that the Host can later read.
- Program downloading. Because the Instruction Closely Coupled Memory is implemented in the same SRAM as the DCCM, the Embedded Memory Interface can be used by the Host to download new program segments for the EC. The Read/Write window would be configured by the Host to point to the beginning of the loadable program region, which could then be loaded by the Host.
- Data exchange. The Read/Write portion of the memory window can be used to contain a communication packet. The Host, by default, “owns” the packet, and can write it at any time. When the Host wishes to communicate with the EC, it sends the EC a command, through the Host-to-EC message facility, to read the packet and perform some operations as a result. When it is completed processing the packet, the EC can inform the Host, either through a message in the EC-to-Host channel or by triggering an event such as an SMI directly. If return results are required, the EC can write the results into the Read/Write region, which the Host can read directly when it is informed that the EC has completed processing. Depending on the command, the operations could entail update of virtual registers in the DCCM, reads of any register in the EC address space, or writes of any register in the EC address space. Because there are two regions that are defined by the base registers, the memory used for the communication packet does not have to be contiguous with a set of virtual registers.

Because there are two Embedded Memory Interface memory regions, the Embedded Memory Interface cannot be used for more than two of these functions at a time. The Host can request that the EC switch from one function to another through the use of the Host-to-EC mailbox register.

The Application ID Register is provided to help software applications track ownership of an Embedded Memory Interface. An application can write the Application ID Register with its Application ID, then immediately read it back. If the read value is not the same as the value written, then another application has ownership of the interface.

Note 1: The protocol used to pass commands back and forth through the Embedded Memory Interface Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Embedded Memory Interface registers to gain access to all of the EC registers.

2: The EC must be awake (i.e., the EC must not be in sleep mode with its clocks gated) in order for the EMI to read or write data in the DCCM/ICCM. System software can insure that the EC is awake when the Host is accessing the EMI with an appropriate protocol using the Host-to-EC mailbox register. Before accessing the memory, the Host sends a “Do Not Sleep” command to the EC through the mailbox register. Writing the register generates an EC interrupt, which wakes the EC. The “Do Not Sleep” command sets a state bit that the EC can check before it issues the Sleep command. When the Host has completed accessing the memory, the Host sends a “Sleep Permitted” command to the EC.

7.5 Registers

The Embedded Memory Interface has its own Logical Device Number, and Base Address as indicated in Table 7-1. The Host LPC I/O addresses for the Embedded Memory Interface are selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through Host Access Configuration Port (see Section 4.5.1, "Host Access Port," on page 55.)

Table 7-2 is a register summary for the Embedded Memory Interface block.

TABLE 7-1: Embedded Memory Interface BASE ADDRESS TABLE

Embedded Memory Interface **Instance**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
EM Interface	10h	FF_4000h

Note: The Host LPC I/O addresses for this instance is selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through the Host Access Configuration Port (see Section 4.5.1, "Host Access Port," on page 55).

The Table 7-2 is a register summary for one instance of the Embedded Memory Interface. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register is accessed through the Host Access Port is via its LDN indicated in Table 7-1 on page 139 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 7-2: Embedded Memory Interface REGISTER SUMMARY

	Host I/O Access			**EC Interface**			
Register Name	**Host I/O Offset**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
HOST-to-EC Mailbox Register	00h	00h	R/W	00h	0	R/W	Note 7-1
EC-to-Host Mailbox Register	01h	01h	R/WC	01h	1	R/WC	Note 7-2
EC Address Register	02h 03h	02h 03h	R/W	02h	2-3	R/W	
EC Data Register	04h 05h 06h 07h	04h 05h 06h 07h	R/W	04h	0-3	R/W	
Interrupt Source Register	08h 09h	08h 09h	Table 7-9	08h	0-1	Table 7-9	
Interrupt Mask Register	0Ah 0Bh	0Ah 0Bh	R/W	0Ah	2-3	R/W	
Application ID Register	0Ch	0Ch	R/W	0Ch	0	R/W	
HOST-to-EC Mailbox Register	-	-	-	100h	0	R/WC	Note 7-1

TABLE 7-2: Embedded Memory Interface REGISTER SUMMARY (CONTINUED)

	Host I/O Access			EC Interface			
Register Name	**Host I/O Offset**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
EC-to-Host Mailbox Register	-	-	-	101h	1	R/W	Note 7-2
Memory Base Address 0 Register	-	-	-	104h	0-3	R/W	
Memory Read Limit 0 Register	-	-	-	108h	0-1	R/W	
Memory Write Limit 0 Register	-	-	-	10Ah	2-3	R/W	
Memory Base Address 1 Register	-	-	-	10Ch	0-3	R/W	
Memory Read Limit 1 Register	-	-	-	110h	0-1	R/W	
Memory Write Limit 1 Register	-	-	-	112h	2-3	R/W	
Interrupt Set Register	-	-	-	114h	0-1	R/W	
Host Clear Enable Register	-	-	-	116h	2-3	R/W	

Note 7-1 Interrupt is cleared when read by the EC.

Note 7-2 Interrupt is cleared when read by the host.

7.5.1 EMBEDDED MEMORY INTERFACE CONTROL REGISTERS

Mailbox Register, HOST-to-EC, and Mailbox Register, EC-to-HOST, are specifically designed to pass commands between the host and the EC (FIGURE 7-1: on page 134). If enabled, these registers can generate interrupts.

When the host performs a write of the HOST-to-EC mailbox register, an interrupt will be generated and seen by the EC if unmasked. When the EC writes the HOST-to-EC mailbox register using the EC-only offset address 100h, it can reset the register to 00h, providing a simple means for the EC to inform the host that an operation has been completed.

When the EC writes the EC-to-HOST mailbox register, an SIRQ event or an event such as SMI or PME may be generated and seen by the host if unmasked. The Host CPU can reset the EC-to-HOST mailbox register to 00h, providing a simple means for the host to inform that EC that an operation has been completed.

PROGRAMMER'S NOTE: The protocol used to pass commands back and forth through the Mailbox Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Mailbox registers to gain access to all of the EC registers.

7.6 Registers

7.6.1 HOST-TO-EC MAILBOX REGISTER

TABLE 7-3: HOST-TO-EC MAILBOX REGISTER

HOST OFFSET	00h			8-Bit			**HOST SIZE**	
EC OFFSET	00h			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	HOST_EC_MBOX[7:0]							

TABLE 7-4: HOST-TO-EC MAILBOX REGISTER

HOST OFFSET	-			-			**HOST SIZE**	
EC OFFSET	100h			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	HOST_EC_MBOX[7:0]							

HOST_EC_MBOX[7:0]

If enabled, an interrupt to the EC marked by the EM_MBX bit in the GIRQ15 Source Register will be generated whenever the Host writes this register. The Host and the EC can read and write this register at offset 000h. The EC can also read this register at offset 100h.

Writes of a 1 to any bit in this register by the EC to this register at offset 100h will cause the bit to be cleared. Writes of a 0 to any bit have no effect.

7.6.2 EC-TO-HOST MAILBOX REGISTER

TABLE 7-5: EC-TO-HOST MAILBOX REGISTER

HOST OFFSET	01h			8-Bit				**HOST SIZE**
EC OFFSET	01h			8-Bit				**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_HOST_MBOX[7:0]							

TABLE 7-6: EC-TO-HOST MAILBOX REGISTER

HOST OFFSET	-			-				**HOST SIZE**
EC OFFSET	101h			8-Bit				**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_HOST_MBOX[7:0]							

EC_HOST_MBOX[7:0]

An EC write to this register will set bit EC_WR in the Interrupt Source Register to ‘1b’. If enabled, setting bit EC_WR to ‘1b’ generates a Host SIRQ event as well was the external EM_Int event. The EC can also read and write this register at offset 101h.

Writes of a 1 to any bit in this register at offset 01h, by the Host or by the EC, will cause the bit to be cleared. Writes of a 0 to any bit have no effect.

7.6.3 EC ADDRESS REGISTER

TABLE 7-7: EC ADDRESS REGISTER

HOST OFFSET	Byte 0: 02h Byte 1: 03h			8-bit			**HOST SIZE**	
EC OFFSET	02h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Region	EC_Address[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_Address[7:2]						Access_Type	

ACCESS_TYPE

This field defines the type of access that occurs when the EC Data Register is read or written.

00: 8-bit access. Any byte read of Byte 0 through Byte 3 in the EC Data Register causes the corresponding byte within the 32-bit double word addressed by EC_Address to be loaded into the byte of EC Data Register and returned by the read. Any byte write to Byte 0 through Byte 3 in the EC Data Register writes the corresponding byte within the 32-bit double word addressed by EC_Address, as well as the byte of the EC Data Register.

01: 16-bit access. A read of Byte 0 in the EC Data Register causes the 16 bits in the DCCM at an offset of EC_Address to be loaded into Byte 0 and Byte 1 of the EC Data Register. The read then returns the contents of Byte 0. A read of Byte 2 in the EC Data Register causes the 16 bits in the DCCM at an offset of EC_Address+2 to be loaded into Byte 2 and Byte 3 of the EC Data Register. The read then returns the contents of Byte 2. A read of Byte 1 or Byte 3 in the EC Data Register return the contents of the register, without any update from the DCCM.

A write of Byte 1 in the EC Data Register causes Bytes 1 and 0 of the EC Data Register to be written into the 16 bits in the DCCM at an offset of EC_Address. A write of Byte 3 in the EC Data Register causes Bytes 3 and 2 of the EC Data Register to be written into the 16 bits in the DCCM at an offset of EC_Address+2. A write of Byte 0 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the DCCM.

10: 32-bit access. A read of Byte 0 in the EC Data Register causes the 32 bits in the DCCM at an offset of EC_Address to be loaded into the entire EC Data Register. The read then returns the contents of Byte 0. A read of Byte 1, Byte 2 or Byte 3 in the EC Data Register returns the contents of the register, without any update from the DCCM.

A write of Byte 3 in the EC Data Register causes the EC Data Register to be written into the 32 bits in the DCCM at an offset of EC_Address. A write of Byte 0, Byte 1 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the DCCM.

11: Auto-increment 32-bit access. This defines a 32-bit access, as in the 10 case. In addition, any read or write of Byte 3 in the EC Data Register causes the EC Address Register to be incremented by 1. That is, the EC_Address field will point to the next 32-bit double word in the DCCM.

EC_ADDRESS[14:2]

This field defines the location in memory that can be read and/or written with the EC Data Register. The address is an offset from the base of the Host-accessible region in the EC DCCM SRAM. The base of the Host-accessible region.

REGION

When this bit is 0, the address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 0 Register. When this bit is 1, the address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 1 Register.

7.6.4 EC DATA REGISTER

TABLE 7-8: EC DATA REGISTER

HOST OFFSET	Byte 0: 04h Byte 1: 05h Byte 2: 06h Byte 3: 07h			8-bit			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data3[7:0]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data2[7:0]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data1[7:0]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data0[7:0]							

DATA

This is a 32-bit register which returns data to the Host from the EC DCCM at the address specified by EC_Address[14:2]. The description of bits Access_Type in the EC Address Register defines which reads and writes from the Host trigger transfers of data between this register and the DCCM.

A write to the EC Data Register when the EC Address Register is in a read-only or a no-access region, as defined by the Memory Base and Limit registers, will update the EC Data Register but memory will not be modified. A read to the EC Data Register when the EC Address Register is in a no-access region, as defined by the Memory Base and Limit

registers, will not trigger a memory read and will not modify the EC Data Register. In auto-increment mode (Access_Type=11b), reads of Byte 3 of the EC Data Register will still trigger increments of the EC Address Register when the address is out of bounds, while writes of Byte 3 will not.

7.6.5 INTERRUPT SOURCE REGISTER

TABLE 7-9: INTERRUPT SOURCE REGISTER

HOST OFFSET	Byte 0: 08h Byte 1: 09h			8-Bit			**HOST SIZE**	
EC OFFSET	08h			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_SWI[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R
BIT NAME	EC_SWI[6:0]							EC_WR

EC_WR

This bit is set autonomously when the EC-to-Host Mailbox Register has been written by the EC at offset 101h. An SIRQ event or an external nEM_Int event to the Host is generated when any bit in this register (EC_WR or any bit in EC_SWI[14:0]) is ‘1b’ and the corresponding bit in the Interrupt Mask Register register is ‘1b’.

This bit is automatically cleared by a read of the EC-to-Host Mailbox Register at offset 01h.

EC_SWI[14:0]

Each bit in this field is cleared when written with a ‘1b’. The ability to clear the bit can be disabled if the corresponding bit in the Host Clear Enable Register is set to ‘0b’.

The EC can generate an interrupt to the Host by setting any bit in this field to ‘1b’. The EC can set bits to ‘1b’ by writing the corresponding bit in the Interrupt Set Register to ‘1b’.

7.6.6 INTERRUPT MASK REGISTER

TABLE 7-10: INTERRUPT MASK REGISTER

HOST OFFSET	Byte 0: 0Ah Byte 1: 0Bh			8-Bit			**HOST SIZE**	
EC OFFSET	0Ah			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_SWI_EN[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_SWI_EN[6:0]							EC_WR_EN

EC_WR_EN

If this bit is ‘1b’, the interrupt generated by bit EC_WR in the Interrupt Source Register is enabled.

EC_SWI_EN[14:0]

Each bit that is set to ‘1b’ in this field enables the generation of and interrupt by the corresponding bit in the EC_SWI[14:0] field in the Interrupt Source Register.

7.6.7 APPLICATION ID REGISTER

TABLE 7-11: APPLICATION ID REGISTER

HOST OFFSET	0Ch			8-Bit			**HOST SIZE**	
EC OFFSET	0Ch			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Application_ID[7:0]							

APPLICATION_ID

When this field is 00h it can be written with any value. When set to a non-zero value, writing that value will clear this register to 00h. When set to a non-zero value, writing any value other than the current contents will have no effect.

7.6.8 MEMORY BASE ADDRESS 0 REGISTER

This register is accessible to the EC only.

TABLE 7-12: MEMORY BASE ADDRESS 0 REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	104h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_0[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_0[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Base_Address_0[7:2]						Reserved	

MEMORY_BASE_ADDRESS_0[23:2]

This register defines the beginning of a region in the Embedded Controller's Data Closely Coupled Memory that is shared between the Host and the EC. The region defined by this base register, Region 0, is used when bit 15 of the EC Address Register is 0. The access will be to a memory location at an offset defined by the contents of the EC Address Register, relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at DCCM_Base_Address + Memory_Base_Address_0[23:2] + EC_Address[14:2].

For example, if Region = 0, the Memory_Base_Address_0[23:2] = 1000h and the EC_Address[14:2] = 20h, then the AHB address of the access will be 80_1020h.

7.6.9 MEMORY READ LIMIT 0 REGISTER

This register is accessible to the EC only.

TABLE 7-13: MEMORY READ LIMIT 0 REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	108h				16-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Read_Limit_0[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Read_Limit_0[7:2]						Reserved	

MEMORY_READ_LIMIT_0[14:2]

Whenever a read of any byte in EC Data Register is attempted, and bit 15 of EC_Address is 0, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Read_Limit_0[14:2] the EC Data Register will be loaded from the DCCM.

7.6.10 MEMORY WRITE LIMIT 0 REGISTER

This register is accessible to the EC only.

TABLE 7-14: MEMORY WRITE LIMIT 0 REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	10Ah				16-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Write_Limit_0[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Write_Limit_0[7:2]						Reserved	

MEMORY_WRITE_LIMIT_0[14:2]

Whenever a write of any byte in EC Data Register is attempted and bit 15 of EC_Address is 0, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_0[14:2] the addressed bytes in the EC Data Register will be written into the DCCM. If EC_Address[14:2] is greater than or equal to Memory_Write_Limit_0[14:2] no writes will take place.

7.6.11 MEMORY BASE ADDRESS 1 REGISTER

This register is accessible to the EC only.

TABLE 7-15: MEMORY BASE ADDRESS 1 REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	10Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_1[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_1[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Base_Address_1[7:2]						Reserved	

MEMORY_BASE_ADDRESS_1[23:2]

This register defines the beginning of a region in the Embedded Controller's Data Closely Coupled Memory that is shared between the Host and the EC. The region defined by this base register, Region 1, is used when bit 15 of the EC Address Register is 1. The access will be to a memory location at an offset defined by the contents of the EC Address Register, relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at DCCM_Base_Address + Memory_Base_Address_1[23:2] + EC_Address[14:2].

For example, if Region = 1, the Memory_Base_Address_1[23:2] = 1000h and the EC_Address[14:2] = 20h, then the AHB address of the access will be 80_1020h.

7.6.12 MEMORY READ LIMIT 1 REGISTER

This register is accessible to the EC only.

TABLE 7-16: MEMORY READ LIMIT 1 REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	110h				16-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Read_Limit_1[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Read_Limit_1[7:2]						Reserved	

MEMORY_READ_LIMIT_1[14:2]

Whenever a read of any byte in EC Data Register is attempted, and bit 15 of EC_Address is 1, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Read_Limit_1[14:2] the EC Data Register will be loaded from the DCCM.

7.6.13 MEMORY WRITE LIMIT 1 REGISTER

This register is accessible to the EC only.

TABLE 7-17: MEMORY WRITE LIMIT 1 REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	112h				16-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Write_Limit_1[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Write_Limit_1[7:2]						Reserved	

MEMORY_WRITE_LIMIT_1[14:2]

Whenever a write of any byte in EC Data Register is attempted and bit 15 of EC_Address is 1, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_1[14:2] the addressed bytes in the EC Data Register will be written into the DCCM. If EC_Address[14:2] is greater than or equal to Memory_Write_Limit_1[14:2] no writes will take place.

7.6.14 INTERRUPT SET REGISTER

This register is accessible to the EC only.

TABLE 7-18: INTERRUPT SET REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	114h			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS
BIT NAME	EC_SWI_Set[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R
BIT NAME	EC_SWI_Set[6:0]							Reserved

EC_SWI_SET[14:0]

This register provides the EC with a means of updating the Interrupt Source Register. Writing a bit in this field with a ‘1b’ sets the corresponding bit in the Interrupt Source Register to ‘1b’. Writing a bit in this field with a ‘0b’ has no effect. Reading this field returns the current contents of the Interrupt Source Register.

7.6.15 HOST CLEAR ENABLE REGISTER

This register is accessible to the EC only.

TABLE 7-19: HOST CLEAR ENABLE REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	116h			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Host_Clr_Enable[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
BIT NAME	Host_Clr_Enable[6:0]							Reserved

HOST_CLR_ENABLE[14:0]

When a bit in this field is ‘0b’, the corresponding bit in the Interrupt Source Register cannot be cleared by writes to the Interrupt Source Register. When a bit in this field is ‘1b’, the corresponding bit in the Interrupt Source Register can be cleared when that register bit is written with a ‘1b’.

These bits allow the EC to control whether the status bits in the Interrupt Source Register are based on an edge or level event.

8.0 VLPC BUS INTERFACE

8.1 General Description

8.1.1 OVERVIEW

The VLPC Bus interface enables communication between either the host processor or the MEC1609/MEC1609i and one or more Companion devices located on a Very Low Pin Count (VLPC) bus. The VLPC Bus is a proprietary interconnect designed to enable communication between a master device, a TPM, and up to three peripheral slaves. The VLPC Bus is fully described in the document *Microchip VLPC Bus*.

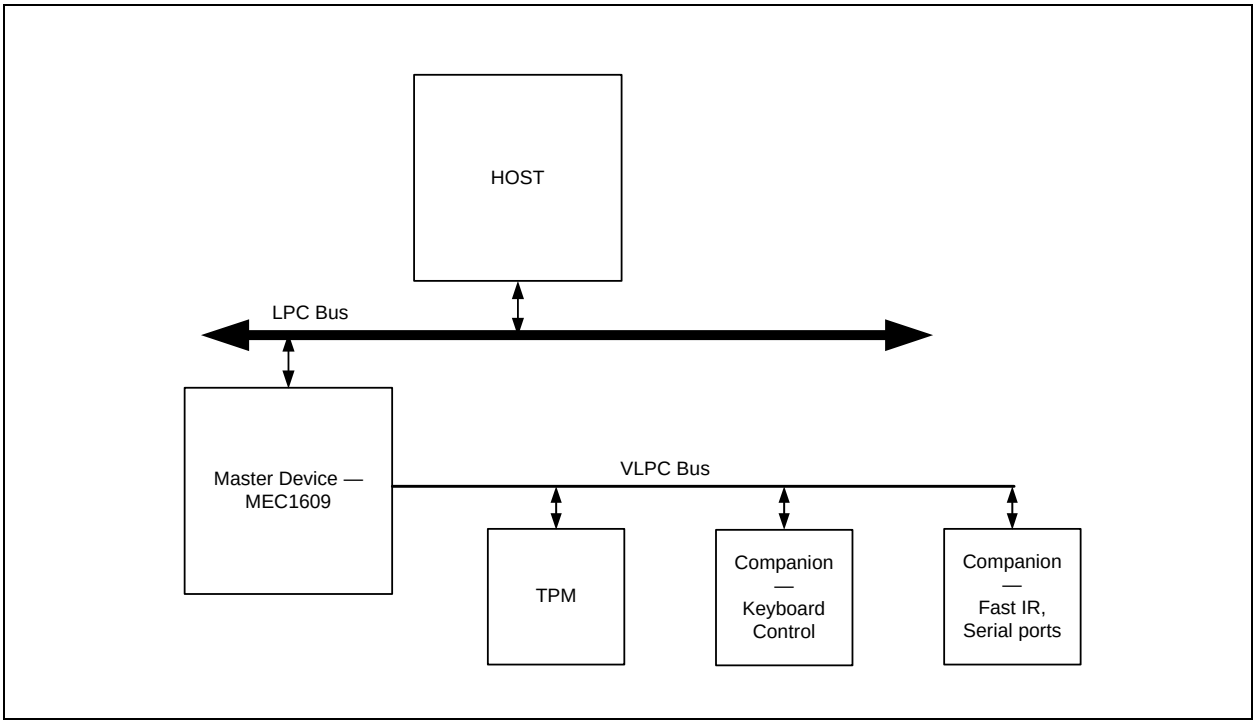
8.1.2 FEATURES

The VLPC Bus Interface block performs the following functions:

- Forwards LPC I/O Reads and I/O Writes from the Host to Companion devices located on the VLPC bus
- Supports 8-bit and 16-bit Reads and Writes from the EC to Companion devices
- Supports LPC DMA transactions from the Host to Companion devices located on the VLPC bus
- Supports interrupts and DMA requests from devices on the VLPC bus targeting either the Host or the EC on the MEC1609/MEC1609i
- Enables access to logical devices in Companion chips from both the Host and the EC on the MEC1609/MEC1609i
- Provides secure Host or EC access to a TPM located on the VLPC bus

8.2 Block Diagram

FIGURE 8-1: VLPC INTERFACE IN MEC1609/MEC1609I



8.3 Power, Clocks and Reset

8.3.1 POWER DOMAIN

This block is powered by VTR.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

8.3.2 CLOCKS

Registers in the VLPC Configuration Block are clocked on the LPC Bus Clock. The VLPC state machine is clocked by MCLK, which is also used to derive V_CLK signal pin function.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

8.3.3 RESET

This block is reset on a nSYS_RST. On reset, the VLPC interface resets to Idle state.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

8.4 Interrupts

The VLPC Bus Interface can generate VLPC_EVT[0:4], VLPC_EVT[5:9], VLPC_EVT[10:14] and VLPC_TPM interrupts when the VLPC events are acquired from companions. The VLPC_EC interrupt reports three VLPC controller-detected conditions reported in the Interface Control Register on page 157.

All sixteen interrupts from the VLPC Bus Interface are level, active high signals and are routed onto the GIRQ21 Source Register on page 284.

When the VLPC bus is idle and V_CLK is off, a Companion device on the VLPC bus can signal the MEC1609/MEC1609i that it requires service by driving V_DATA low. If this occurs while the MEC1609/MEC1609i ring oscillator is running, V_CLK is restarted, the Attention bit in the Interface Control Register on page 157 is set and the VLPC_EC interrupt is reported the GIRQ21 Source Register.

The VLPC_EC interrupt is not an EC wake-capable interrupt. If a Companion signals the MEC1609/MEC1609i by driving V_DATA low while the ring oscillator is stopped, V_CLK is not restarted and no VLPC_EC interrupt event occurs. The “VLPC bus wakeup event” can be detected, when the ring oscillator is running or stopped, by the V_DATA pin signal edge detection interrupt and wake event. The V_DATA wakeup event is controlled by its associated Pin Control Register. (See Section 22.0, "GPIO Interface," on page 329. The “VLPC bus wakeup event” detection Wake-up event is routed to the V_DATA bit in the GIRQ21 Source Register on page 284. When the V_DATA pin is configured for edge detection and wake events, V_CLK will restart when a Companion signals the MEC1609/MEC1609i for service.

8.4.1 VLPC Bus Interface SIRQ ROUTING

The VLPC Bus Interface can generate SIRQ events for the companion events. See Section 4.0, "Logical Device Configuration", Section 4.8, "SERIRQ Interrupts," on page 62

8.4.2 VLPC BUS SIGNALS

The VLPC bus consists of three signals, shown in Table 8-1, "VLPC Bus Signal List":

TABLE 8-1: VLPC BUS SIGNAL LIST

Signal	Description
V_CLK	Clock
V_FRAME	Frame
V_DATA	Data

See Section 2.0, "Pin Configuration," on page 8 for details on these pins, Table 2-19 on page 19.

8.4.3 VLPC CONNECTIVITY

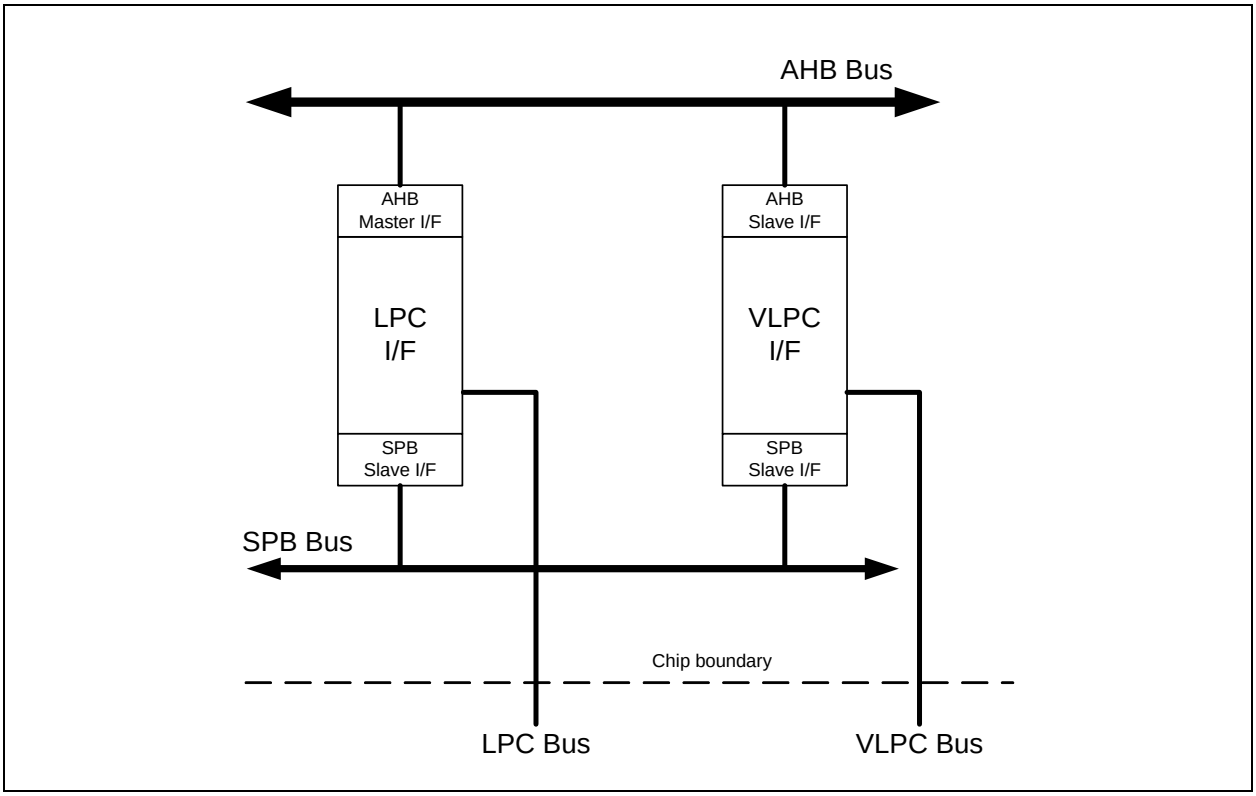
All registers in Companion devices on the VLPC bus are located in the MEC1609/MEC1609i AHB address space. The VLPC Bus interface is connected to the MEC1609/MEC1609i AHB bus through an AHB slave interface, so the EC can access all addresses in the VLPC address space through AHB reads and writes. The VLPC bus supports both 8-bit and 16-bit reads and writes, although individual Companion devices may or may not be able to support 16-bit accesses. The mapping of AHB address to VLPC addresses is summarized in Table 8-2, "AHB Address Mapping to VLPC Bus".

TABLE 8-2: AHB ADDRESS MAPPING TO VLPC BUS

AHB Address	VLPC Bus Address	VLPC Companion Address
FE_0000h - FE_3FFFh	I/O space: 0h - 3FFFh	Companion 0: 0h - 3FFFh
FE_4000h - FE_7FFFh	I/O space: 4000h - 7FFFh	Companion 1: 0h - 3FFFh
FE_8000h - FE_BFFFh	I/O space: 8000h - BFFFh	Companion 2: 0h - 3FFFh
FE_C000h - FE_DFFFh	I/O space: C000h - DFFFh	Global Read/Write Registers Companions 0, 1, 2 C000h - DFFFh
FE_E000h - FE_FFFFh	I/O space: E000h - FFFFh	Global Write-only Registers Companions 0, 1, 2 E000 h- FFFFh
FD_0000h - FD_4FFFh	TPM space: 0h - 4FFFh	TPM 0h - 4FFFh
FD_5000h - FD_FFFFh	Reserved	Reserved

LPC-VLPC interconnections are shown in Figure 8-2, "LPC-VLPC Interconnection".

FIGURE 8-2: LPC-VLPC INTERCONNECTION



8.4.3.1 VLPC Bus Arbitration

There are two potential sources for a VLPC bus transaction:

- A bus transaction on the LPC bus
- A read or write to the VLPC Bus address space by the EC

The priority order for transactions is:

1. Completion of the transaction currently on the VLPC Bus
2. An I/O or TPM Read or Write on the LPC bus that maps to VLPC Bus address space
3. A Read or Write from the EC to an AHB address in the range FD_0000h through FF_FFFFh

Because the LPC bus has priority over the EC, if an EC access in the range FD_0000h through FF_FFFFh arrives after an LPC transaction begins on the LPC bus but before the LPC Controller has translated the LPC address, the EC transaction will be stalled. If the LPC Controller claims the LPC address, the EC transaction will be stalled until the LPC transaction completes.

8.4.4 VLPC CONFIGURATION BLOCK

The registers listed here control the operation of the VLPC Bus.

The VLPC Bus Interface has a VLPC Configuration Block which has its own Logical Device Number, and Base Address as indicated in Table 8-3. The Host LPC I/O addresses for the VLPC Configuration Block are selected via Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through Host Access Configuration Port (see Section 4.5.1, "Host Access Port," on page 55).

TABLE 8-3: VLPC Configuration Block BASE ADDRESS TABLE

VLPC Configuration Block	Logical Device Number (from Table 3-2 on page 48)	AHB Base Address
VLPC Interface	Dh	FF_3400h

Note: The Host LPC I/O addresses for this instance are selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through the Host Access Configuration Port (see Section 4.5.1, "Host Access Port," on page 55).

Table 8-4 is a register summary for the VLPC Configuration Block. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port address via its LND indicated in Table 8-3 on page 156 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 8-4: VLPC Bus Interface REGISTER SUMMARY

	Host Access			EC Access			
Register Name	**Host Config. Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Activate Register	30	330h	R/W	330h	0	R/W	
Interface Control Register	F0h	3F0h	R/W	3F0h	0	R/W	
	F1h	3F1h	R/W	3F1h	1	R/W	
Error Address Register	F4h	3F4h	R/W	3F4h	0	R/W	
	F5h	3F5h	R/WC	3F5h	1	R/W	
VLPC Clock Control Register	F8h	3F8h	R/W	3F8h	0	R/W	

8.4.4.1 Interface Control

TABLE 8-5: INTERFACE CONTROL REGISTER

HOST CONFIG. INDEX	BYTE0: F0h BYTE1: F1h			8-bit			**HOST SIZE**	
EC OFFSET	3F0h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**…**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/WC	R/WC	R/WC
BIT NAME	LPC_ Berr	EC_ Berr	Attn_ Enable	Err_I_ Enable	TO_I_ Enable	Attention	Trans_ Err	Trans_ Timeout
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R/W	R/W	R/W	R	W	R/W
EC TYPE	R	R	R/W	R/W	R/W	R	W	R/W
BIT NAME	Reserved	CSDA	Poll_ TPM/0	Poll_ 1/2	Power_ Down	V_CLK	Soft_ Reset	Write_ Protect

WRITE_PROTECT

When this bit is 0, all registers in the Configuration Register address range are writable by both the Host and the EC. When this bit is 1, all registers in the range are only writable by the EC.

SOFT_RESET

Writing a 1 to this bit resets the VLPC Bus Interface, resetting the bus interface state machine. This bit is write-only and returns 0 on reads.

V_CLK

This bit reports the state of V_CLK. If this bit is 0, then the clock is inactive and the VLPC Bus is in the idle, low-power state. When this bit is 1, then the VLPC Bus is active and running normally. This bit is read-only.

POWER_DOWN

When this bit is 1, the VLPC bus will issue the *Clock Stop Power Down Broadcast* transaction when there are no data transactions pending. When this bit is 0, the VLPC bus will issue the *Clock Stop Broadcast* transaction when there are no data transactions pending.

POLL_1/2

When this bit is 1, the Companion 1/2 Event Polling is enabled. When this bit is 0, Companion 1/2 Event Polling is suppressed.

This bit is set and cleared by software.

APPLICATION NOTE: Software should insure that either a Companion at ID 1 or a Companion at ID 2 is present before setting this bit.

POLL_TPM/0

When this bit is 1, the Companion TPM/0 Event Polling is enabled. When this bit is 0, Companion TPM/0 Event Polling is suppressed.

This bit is set and cleared by software.

APPLICATION NOTE: Software should insure that either a Companion at ID 0 or a TPM is present before setting this bit.

CSDA

Clock Stop Disable. When this bit is 1, the automatic *Clock Stop Broadcast* transactions are suppressed and V_CLK continues to run after every transaction. This bit defaults to 0.

TRANS_TIMEOUT

When this bit is 1, a bus timeout occurred on the VLPC Bus. This bit is sticky: once set by a bus timeout, it remains set. It is cleared by writing a 1 to this bit.

TRANS_ERR

When this bit is 1, a VLPC bus transaction was terminated with an Error return (a RESPONSE code of '1100b'). This bit is sticky: once set by a bus error, it remains set. It is cleared by writing a 1 to this bit.

ATTENTION

When this bit is 1, a Slave Device has signaled the VLPC Bus that an event occurred in the device while the VLPC Bus clock was stopped. This bit is sticky: once set by an attention event, it remains set. It is cleared by writing a 1 to this bit.

TO_I_ENABLE

When this bit is 1, a VLPC bus timeout will generate an interrupt to the Embedded Controller on the Master Device. When this bit is a 0, a VLPC bus timeout will be recorded in Trans_Timeout but otherwise ignored.

ERR_I_ENABLE

When this bit is 1, a VLPC bus transaction terminated with an Error return (a RESPONSE code of '1100b') will generate an interrupt to the Embedded Controller. When this bit is a 0, a VLPC bus timeout will be recorded in Trans_Err but otherwise ignored.

ATTN_ENABLE

When this bit is 1, a 1 in the Attention bit will generate an interrupt to the Embedded Controller on the Master Device. When this bit is a 0, the Attention events will be recorded in the Attention bit but will not generate an interrupt to the Embedded Controller. If Event Polling is enabled by bits Poll_TPM/0 and Poll_1/2 in the Interface Control register, Event Polls will still take place, and an interrupt to the Embedded Controller could be generated by a Slave Device if it is so configured in the *Event Config registers*.

EC_BERR

When this bit is 1, a VLPC bus timeout or an Error RESPONSE on a VLPC bus transaction initiated in response to a request from the Embedded Controller will cause a bus error on the Embedded Controller. When VLPC Bus Write transactions are terminated with an error the VLPC Bus controller will silently terminate the Embedded Controller without a bus error. Independent of this bit, a VLPC Bus transaction that was terminated with an error will return FFh on 8-bit Reads, FFFFh on 16-bit Reads.

LPC_BERR

When this bit is a 1, a VLPC bus timeout or an Error RESPONSE on a VLPC bus transaction initiated in response to a request from the LPC bus will terminate the LPC transaction with an ERROR SYNC. When this bit is a 0, the LPC transaction will be terminated normally even if an error occurred. Independent of this bit, a VLPC Bus transaction that was terminated with an error will return FFh on 8-bit Reads, FFFFh on 16-bit Reads.

8.4.4.2 Error Address

TABLE 8-6: ERROR ADDRESS REGISTER

HOST CONFIG. INDEX	BYTE0: F4h BYTE1: F5h			8-bit			**HOST SIZE**	
EC OFFSET	3F4h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**…**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	Error Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	Error Address[7:0]							

ERROR ADDRESS

The 16-bit address that records the first address that caused an error which set either the Trans_Timeout bit or the Trans_Err bit in the Interface Control Register. Once an error occurs it is not updated until written by software. It is cleared and unlocked on writes. The register locks even if an error occurs on address 00h.

8.4.4.3 VLPC Clock Control Register

TABLE 8-7: VLPC CLOCK CONTROL REGISTER

HOST CONFIG. INDEX	BYTE0: F8h					8-bit	HOST SIZE	
EC OFFSET	3F8h					32-bit	EC SIZE	
POWER	VTR					0000_0000h	nSYS_RST DEFAULT	
BUS	LPC SPB							
BYTE[3:1] BIT	D31	D30	D29	…		D10	D9	D8
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
HOST TYPE	R	R	R	R	R	R	R/W	R/W
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						Clock_Divider	

CLOCK_DIVIDER

The VLPC bus clock can be set to four values based on this field. The values are:

0: 64MHz

1: 32MHz

2: 16MHz

3: 8MHz

Writes to this register do not immediately change the VLPC Bus clock, which can only be changed when the clock is stopped. Writes to this register write to a buffer, which is copied into the VLPC Clock Control Register only when the V_CLK bit in the Interface Control Register is 0 (that is, when the VLPC Bus clock is stopped).

8.4.4.4 Activate

TABLE 8-8: ACTIVATE REGISTER

HOST CONFIG. INDEX	BYTE0: 30h					8-bit	HOST SIZE	
EC OFFSET	330h					32-bit	EC SIZE	
POWER	VTR					00b	nSYS_RST DEFAULT	
BUS	LPC SPB							
BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the VLPC logical device is powered and functional. When this bit is 0, the VLPC logical device is powered down and inactive.

Note:	The VLPC pads should be configured for VLPC use before the VLPC interface is activated.

9.0 ACPI EMBEDDED CONTROLLER INTERFACE

9.1 General Description

The ACPI Embedded Controller Interface is a Host/EC Message Interface. The ACPI defines the standard hardware and software communications interface between the OS and an embedded controller. This interface allows the OS to support a standard driver that can directly communicate with the embedded controller, allowing other drivers within the system to communicate with and use the EC resources; for example, Smart Battery and AML code

The ACPI Embedded Controller Interface also provides a four byte data interface which is a superset of the standard ACPI Embedded Controller Interface one byte data interface. The ACPI Embedded Controller Interface defaults to the standard one byte interface.

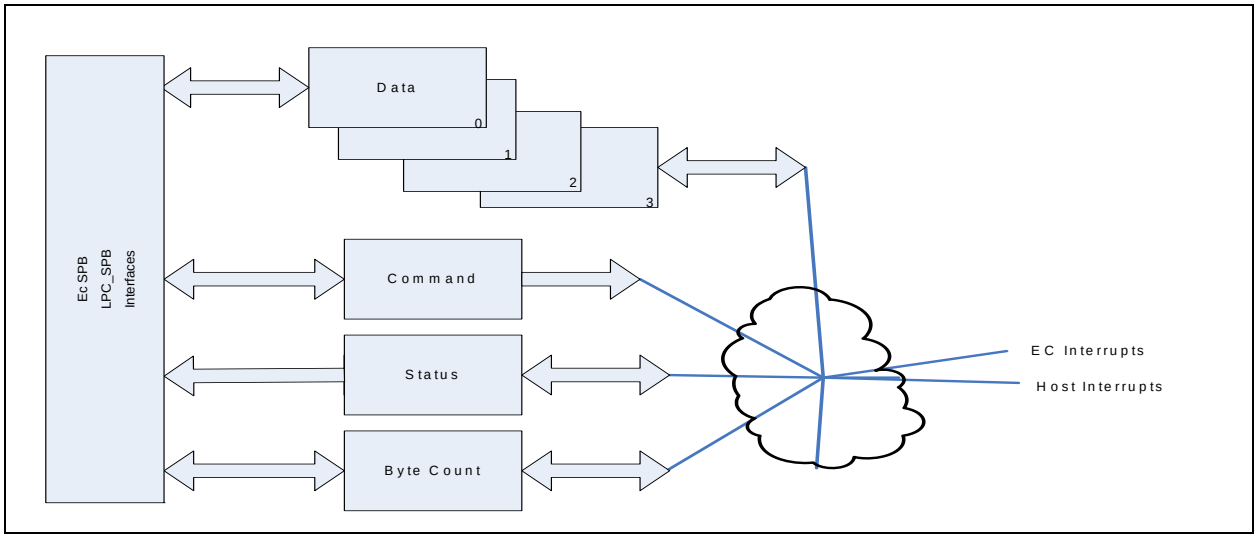
Note:	There is no VCC emulation in hardware for this interface.

The ACPI Embedded Controller Interface features:

- Four Independent Interfaces
- Considered as Channel
- Each Channel is associated with a Logical Device Number
- Standard one data byte interface
- Four byte interface superset

9.1.1 BLOCK DIAGRAM

FIGURE 9-1: ACPI Embedded Controller Interface BLOCK DIAGRAM (SINGLE CHANNEL)



9.2 Power, Clocks and Reset

9.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

9.2.2 CLOCKS

This block has one clock input, the LPC Bus Clock.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

9.2.3 RESET

This block is reset on a nSYS_RST.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

9.3 Interrupts

Each ACPI Embedded Controller Interface can generate an EC_OBF interrupt when the OBF bit in the EC_Cx_STATUS Register is cleared (falling edge sensitive) and an EC_IBF interrupt when the IBF bit in the EC_Cx_STATUS Register is set (active high, level sensitive). These interrupt sources are routed to the GIRQ20 Source Register.

Note 9-1 An SCI or SMI event can be tracked by Host/EC firmware via the SCI_EVT & SMI_EVT Bits in the EC_Cx_STATUS Register on page 164; however, the ACPI Embedded Controller Interface does not generate a SCI or SMI events directly. SMI events are generated in the SMI Interrupt Source Register on page 199 of the MailBox Register Interface. SCI event are generated by the ACPI PM1 Block Interface on page 185. (See Section 11.3.1, "SCI Interrupts to the Host," on page 185.)

9.4 Registers

There are four instances of ACPI Embedded Controller Interface block implemented in the MEC1609/MEC1609i enumerated as 0, 1, 2 and 3.

Each instance of the ACPI Embedded Controller Interface has its own Logical Device Number, and Base Address as indicated in Table 9-1.

TABLE 9-1: ACPI Embedded Controller Interface BASE ADDRESS TABLE

ACPI Embedded Controller Interface **Instance**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
ACPI_EC 0	2h	FF_0800h
ACPI_EC 1	3h	FF_0C00h
ACPI_EC 2	4h	FF_1000h
ACPI_EC 3	5h	FF_1400h

Note 9-2 The Host LPC I/O addresses for each instance is selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through the Host Access Configuration Port. (See Section 4.5.1, "Host Access Port," on page 55.)

The Table 9-2 is a register summary for one instance of the ACPI Embedded Controller Interface. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port is via its LDN indicated in Table 9-1 on page 163 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 9-2: ACPI Embedded Controller Interface REGISTER SUMMARY

	Host I/O Access			**CMD** Note 9-3	**EC Interface**			
Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**		**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
EC_Cx_DATA 0 Register	00h	00h	R/W	0	100h	0	R/W	
EC_Cx_DATA 1 Register	01h	01h	R/W	0		1	R/W	
EC_Cx_DATA 2 Register	02h	02h	R/W	0		2	R/W	
EC_Cx_DATA 3 Register	03h	03h	R/W	0		3	R/W	

TABLE 9-2: ACPI Embedded Controller Interface REGISTER SUMMARY (CONTINUED)

Register Name	Host I/O Access			CMD Note 9-3	EC Interface			Notes
	Host I/O Index	SPB Offset	Host Type		SPB Offset	Byte Lane	EC Type	
EC_Cx_COMMAND Register	04h	04h	W	1	100h	0	R	Note 9-4
EC_Cx_STATUS Register	04h	04h	R	0	104h	0	R/W	
EC_Cx_Byte Count Register	05h	05h	R/W	0	105h	0	R/W	
Reserved	06h	06h	R	n/a	106h	0	R	Note 9-5
Reserved	07h	07h	R	n/a	107h	0	R	

Note 9-3 CMD is bit D3 in the EC_Cx_STATUS Register.

Note 9-4 See Section 9.4.3, "EC_Cx_COMMAND Register," on page 166 for description of host access and data flow to EC.

Note 9-5 This reserved register is claimed by the block. Reads return data value of 00h and writes have no effect.

9.4.1 RUN-TIME REGISTERS

ACPI Embedded Controller Interface contains three registers: EC_Cx_COMMAND Register, EC_Cx_STATUS Register, and EC_Cx_DATA. The standard ACPI Embedded Controller Interface registers occupy two addresses in the Host I/O space (Table 9-2).

The EC_Cx_DATA and EC_Cx_COMMAND registers appear as a single 8-bit data register in the EC. The CMD bit in the EC_Cx_STATUS register is used by the EC to discriminate commands from data written by the host to the ECI. CMD is controlled by hardware: host writes to the EC_Cx_DATA register clear the CMD bit; host writes to the EC_Cx_COMMAND register set the CMD bit.

9.4.2 EC_CX_STATUS REGISTER

The EC_Cx_STATUS Register indicates the state of the Embedded Controller Interface of Channel x. To the host, the EC_Cx_STATUS Register is read-only. To the EC, some bits in the EC_Cx_STATUS Register are read-only (Table 9-3). These bits are controlled by hardware. The EC software controlled bits in the EC_Cx_STATUS Register are read/write.

TABLE 9-3: EC_CX_STATUS REGISTER

HOST I/O INDEX	04h			8-bit			**HOST SIZE**	
EC OFFSET	104h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R							
EC TYPE	R/W	R/W	R/W	R/W	R	R/W	R	R
BIT NAME	UD	SMI_EVT	SCI_EVT	BURST	CMD	UD	IBF	OBF

Note 9-6 The UD bits are User-Defined. UD bits are maintained by EC software, only.

APPLICATION NOTE: the IBF and OBFbits are not cleared (‘0’) by hardware when VCC_PWRGD is asserted or when the LPC interface powers down; for example, following system changes state S3->S0, S5->S0, G3-> S0. To clear the IBF bit in firmware, read offset 100h; to clear the OBF bit in firmware, read offset 000h.

OBF

The Output Buffer Full (OBF) flag is set when the EC writes a byte of data into the data port (EC_Cx_DATA 0 Register).

Once the host reads the status byte and sees the OBF flag set, the host reads the data port to get the byte of data that the EC has written.

Once the host reads the data, the OBF flag is automatically cleared by hardware. An EC_OBF interrupt signals the EC that the data has been read by the host and the EC is free to write more data to the EC_Cx_DATA 0 Register register.

The EC_OBF interrupt is generated whenever the OBF bit in the EC_Cx_STATUS Register register is reset. See Section 9.3, "Interrupts," on page 163.

IBF

The Input Buffer Full (IBF) flag is set when the host has written a byte of data to the command or data port.

An EC_IBF interrupt signals the EC that there is data available. Once the EC reads the status byte and sees the IBF flag set, the EC reads the data port to get the byte of data that the host has written.

Once the EC reads the data, the IBF flag is automatically cleared by hardware. The EC must then generate a software interrupt to alert the host that the data has been read and that the host is free to write more data to the ECI as needed. See Section 9.3, "Interrupts," on page 163.

CMD

The CMD bit is set when the EC_Cx_DATA 0 Register register contains a command byte; the CMD bit is cleared when the EC_Cx_DATA 0 Register contains a data byte.

The CMD bit is controlled by hardware: host writes to the EC_Cx_DATA 0 Register register clear CMD; host writes to the EC_Cx_COMMAND Register set CMD.

The CMD bit allows the embedded controller to differentiate the start of a command sequence from a data byte write operation.

BURST

The BURST bit is set when the EC is in Burst Mode for polled command processing; the BURST bit is cleared when the EC is in Normal mode for interrupt-driven command processing.

The BURST bit is an EC-maintained software flag that indicates the embedded controller has received the Burst Enable command from the host, has halted normal processing, and is waiting for a series of commands to be sent from the host. Burst Mode allows the OS or system management handler to quickly read and write several bytes of data at a time without the overhead of SCIs between commands.

Note 9-7 The BURST bit is maintained by EC software, only.

SCI_EVT

The SCI Event flag is set by software when an SCI event is pending; i.e., the EC is requesting an SCI query; SCI Event flag is clear when no SCI events are pending.

The SCI Event bit is an EC-maintained software flag that is set when the embedded controller has detected an internal event that requires operating system attention. The EC sets SCI Event before generating an SCI to the OS.

Note 9-8 The SCI Event bit is maintained by EC software, only. See Note 9-1 on page 163.

SMI_EVT

The SMI Event flag is set when an SMI event is pending; i.e., the EC is requesting an SMI query; SMI Event flag is cleared when no SMI events are pending.

The SMI Event bit is an EC-maintained software flag that is set when the embedded controller has detected an internal event that requires system management interrupt handler attention. The EC sets SMI Event before generating an SMI.

Note 9-9 The SMI Event flag bit is maintained by EC software, only. See Note 9-1 on page 163.

9.4.3 EC_CX_COMMAND REGISTER

The EC_Cx_COMMAND Register is a write-only register that allows the host to issue commands to the embedded controller.

Writes to the EC_Cx_COMMAND Register are latched in the EC data register and the IBF is set in the EC_Cx_STATUS Register. Writes to the EC_Cx_COMMAND Register also set the CMD bit in the EC_Cx_STATUS Register.

9.4.4 EC_CX_DATA 0 REGISTER

The EC_Cx_DATA 0 Register is a read/write register that allows the host to issue data arguments to the embedded controller and allows the OS to read data returned by the embedded controller.

9.5 Non Legacy Operation

This mode uses three additional Data Registers: EC_Cx_DATA 1 Register, EC_Cx_DATA 2 Register, and EC_Cx_-DATA 3 Register to transfer data to and from the host.

The EC_Cx_Byte Count Register controls the operation of the IBF and OBF; including, the EC_IBF and EC_OBF interrupts. Bits[3:0] of the register contain the Bytes Written flags. These are set when data is written to the corresponding data register and cleared when the corresponding data register is read.

Table 9-5, “Control of IBF,” on page 167 and Table 9-6, “Control of OBF,” on page 168 show an example data flow and the effect on the CMD, IBF, OBF bits and the Bytes Written field.

Host writes to the Command Register are limited to one byte only, regardless of both legacy and non legacy operation.

9.5.1 BYTE COUNT REGISTER

TABLE 9-4: EC_CX_BYTE COUNT REGISTER

HOST OFFSET	05h			8-bit			**HOST SIZE**	
EC OFFSET	105h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R	R	R			
EC TYPE	R/W	R/W	R	R	R			
BIT NAME	Bytes Used		Reserved		Data Byte 3 Written	Data Byte 2 Written	Data Byte 1 Written	Data Byte 0 Written

BYTES USED

Bits[7:6] in the Byte Count Register, the Bytes Used field, indicates how many data bytes are to be transferred. This field defaults to 00h, which supports legacy mode. The values for the Bytes Used bits are:

00 Byte 0 (default legacy)

01 Bytes 0 and 1

10 Bytes 0,1, and 2

11 Bytes 0,1,2, and 3

Note 9-10 All other values are reserved and resort to the default value.

Note 9-11 Host writes to the Command Register are limited to one byte regardless of the value in the Byte Used field in the EC_Cx_Byte Count Register.

BYTES WRITTEN

Bits[3:0], the Bytes Written field, indicates how many bytes have been transferred. This may be used to insure completion of expected data transfers.

TABLE 9-5: CONTROL OF IBF

	CMD	Data Byte 0 Written (Bit[0])	Data Byte 1 Written (Bit[1])	Data Byte 2 Written (Bit[2])	Data Byte 3 Written (Bit[3])	IBF	Bytes Used
Host Writes CMD	↑	↑	0	0	0	↑	00b Legacy
EC Reads CMD	1	↓	0	0	0	↓	
Host Writes Data 0	↓	↑	0	0	0	↑	
EC Reads Data 0	0	↓	0	0	0	↓	
Host Writes CMD	↑	↑	0	0	0	↑	01b
EC Reads CMD	1	↓	0	0	0	↓	
Host Writes Data 0	↓	↑	0	0	0	0	
Host Writes Data 1	0	1	↑	0	0	↑	
EC Reads Data 0	0	↓	1	0	0	1	
EC Reads Data 1	0	0	↓	0	0	↓	
Host Writes CMD	↑	↑	0	0	0	↑	10b
EC Reads CMD	1	↓	0	0	0	↓	
Host Writes Data 0	↓	↑	0	0	0	0	
Host Writes Data 1	0	1	↑	0	0	0	
Host Writes Data 2	0	1	1	↑	0	↑	
EC Reads Data 0	0	↓	1	1	0	1	
EC Reads Data 1	0	0	↓	1	0	1	
EC Reads Data 2	0	0	0	↓	0	↓	

TABLE 9-5: CONTROL OF IBF (CONTINUED)

	CMD	Data Byte 0 Written (Bit[0])	Data Byte 1 Written (Bit[1])	Data Byte 2 Written (Bit[2])	Data Byte 3 Written (Bit[3])	IBF	Bytes Used
Host Writes CMD	↑	↑	0	0	0	↑	11b
EC Reads CMD	1	↓	0	0	0	↓	
Host Writes Data 0	↓	↑	0	0	0	0	
Host Writes Data 1	0	1	↑	0	0	0	
Host Writes Data 2	0	1	1	↑	0	0	
Host Writes Data 3	0	1	1	1	↑	↑	
EC Reads Data 0	0	↓	1	1	1	1	
EC Reads Data 1	0	0	↓	1	1	1	
EC Reads Data 2	0	0	0	↓	1	1	
EC Reads Data 3	0	0	0	0	↓	↓	

TABLE 9-6: CONTROL OF OBF

	CMD	Data Byte 0 Written (Bit[0])	Data Byte 1 Written (Bit[1])	Data Byte 2 Written (Bit[2])	Data Byte 3 Written (Bit[3])	OBF	Bytes Used
EC Writes Data 0	0	↑	0	0	0	↑	00b (Legacy)
Host Reads Data 0	0	↓	0	0	0	↓	
EC Writes Data 0	0	↑	0	0	0	0	01b
EC Writes Data 1	0	1	↑	0	0	↑	
Host Reads Data 0	0	↓	1	0	0	1	
Host Reads Data 1	0	0	↓	0	0	↓	
EC Writes Data 0	0	↑	0	0	0	0	10b
EC Writes Data 1	0	1	↑	0	0	0	
EC Writes Data 2	0	1	1	↑	0	↑	
Host Reads Data 0	0	↓	1	1	0	1	
Host Reads Data 1	0	0	↓	1	0	1	
Host Reads Data 2	0	0	0	↓	0	↓	

TABLE 9-6: CONTROL OF OBF (CONTINUED)

	CMD	Data Byte 0 Written (Bit[0])	Data Byte 1 Written (Bit[1])	Data Byte 2 Written (Bit[2])	Data Byte 3 Written (Bit[3])	OBF	Bytes Used
EC Writes Data 0	0	↑	0	0	0	0	11b
EC Writes Data 1	0	1	↑	0	0	0	
EC Writes Data 2	0	1	1	↑	0	0	
EC Writes Data 3	0	1	1	1	↑	↑	
Host Reads Data 0	0	↓	1	1	1	1	
Host Reads Data 1	0	0	↓	1	1	1	
Host Reads Data 2	0	0	0	↓	1	1	
Host Reads Data 3	0	0	0	0	↓	↓	

9.5.2 EC_CX_DATA 1 REGISTER

The EC_Cx_DATA 1 Register is a read/write register that allows the host to issue data arguments to the embedded controller and allows the OS to read data returned by the embedded controller.

9.5.3 EC_CX_DATA 2 REGISTER

The EC_Cx_DATA 2 Register is a read/write register that allows the host to issue data arguments to the embedded controller and allows the OS to read data returned by the embedded controller.

9.5.4 EC_CX_DATA 3 REGISTER

The EC_Cx_DATA 3 Register is a read/write register that allows the host to issue data arguments to the embedded controller and allows the OS to read data returned by the embedded controller.

10.0 8042 EMULATED KEYBOARD CONTROLLER

10.1 General Description

The MEC1609/MEC1609i keyboard controller uses the EC to produce a superset of the features provided by the industry-standard 8042 keyboard controller. The 8042 Emulated Keyboard Controller is Host/EC Message Interface with hardware assists to emulate 8042 behavior and provide Legacy GATEA20 support.

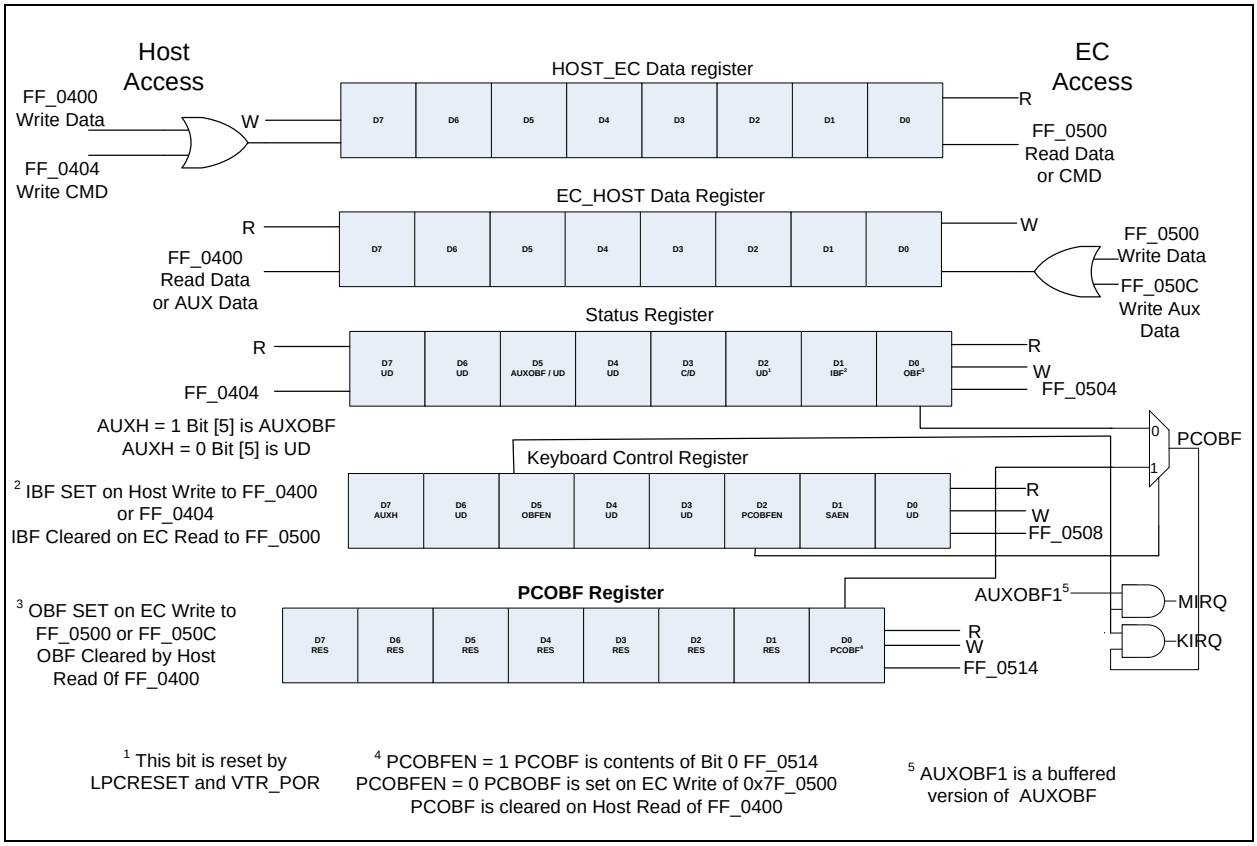
Note:	There is no VCC emulation in hardware for this interface.

10.1.1 FEATURES

- Legacy Keyboard Support
- Emulated 8042 Operation
- Port 92 Legacy A20M Support

10.1.2 BLOCK DIAGRAM

FIGURE 10-1: BLOCK DIAGRAM OF 8042 Emulated Keyboard Controller



10.1.3 BLOCK DIAGRAM SIGNAL LIST

TABLE 10-1: 8042 Emulated Keyboard Controller SIGNAL LIST

Signal Name	Direction	Description
EC_IBF	Output	Interrupt generated by the host writing either data or command to the data register
EC_OBF	Output	Interrupt generated by the host reading either data or aux data from the data register
KIRQ	Output	Routed to the Host SIRQ
MIRQ	Output	Routed to the Host SIRQ
KBRST	Output	Routed to Pin Function

10.2 Power, Clocks and Reset

10.2.1 POWER DOMAIN

This block is powered by VTR.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

10.2.2 CLOCKS

This block has four clock inputs: LPC Bus Clock, 1MHZ (MCLK_DIV32_EN) clock source. LPC Bus Clock is used to the accessible and clock the registers in this block. The 1MHz Host Clock Domain is used to clock the counter in the CPU_RESET circuitry.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

10.3 Power On Reset

This block is reset on a nSYS_RST and nSIO_RESET. On a reset, all Register are reset to 00h and the state machines are set to idle.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

10.4 Interrupts

The 8042 Emulated Keyboard Controller can generate a KBD_OBF interrupt when the OBF bit in the Keyboard Status Read Register is cleared (falling edge sensitive) and a KBD_IBF interrupt when the IBF bit in the Keyboard Status Read Register is set (active high, level sensitive). These interrupt sources are routed to the GIRQ19 Source Register.

10.4.1 8042 EMULATED KEYBOARD CONTROLLER (LDN 1H) SIRQ ROUTING

The 8042 Emulated Keyboard Controller can generate two SIRQ events for the EC-to-HOST EC events:KIRQ & MIRQ. For the KIRQ interrupt the Interrupt Configuration Register, SELECT on page 64 is cleared to ‘0’ and for MIRQ the Interrupt Configuration Register, SELECT is set to’1’.

See Logical Device Configuration, Section 4.8, "SERIRQ Interrupts," on page 62.

10.5 Instance Description

There are two blocks defined in this chapter: 8042 MSG Interface and the Port92-Legacy. The MEC1609/MEC1609i has one instance of each block.

10.6 Registers

The 8042 Emulated Keyboard Controller has a 8042 MSG Interface block which has its own Logical Device Number, and Base Address as indicated in Table 10-2. The Host LPC I/O addresses for the 8042 MSG Interface is selected via Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through Host Access Configuration Port. (See Section 4.5.1, "Host Access Port," on page 55.)

The 8042 Emulated Keyboard Controller also has a Port92-Legacy block which has a separate Logical Device Number and Base Address Register as indicated in Table 10-2. The Base Address Register for the Port92-Legacy has only one writable bit, the Valid Bit, since the only I/O accessible Register has a fixed address.

Table 10-3 is a register summary for the 8042 MSG Interface block and Table 10-14 on page 178 is a register summary for the Port92-Legacy block.

TABLE 10-2: 8042 Emulated Keyboard Controller BASE ADDRESS TABLE

8042 Emulated Keyboard Controller **Blocks**	**LDN from (**Table 3-2 on page 48**)**	**AHB Base Address**
8042 MSG Interface	1h	FF_0400h
Port92-Legacy	8h	FF_2000h

Note 10-1 The Host LPC I/O addresses for this instance is selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through the Host Access Configuration Port. (See Section 4.5.1, "Host Access Port," on page 55.)

Table 10-3 is a register summary for the 8042 MSG Interface block. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port is via its LND indicated in Table 10-2 on page 172 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 10-3: 8042 MSG Interface REGISTER SUMMARY

	Host I/O Access			**CMD** Note 10-2	**EC Interface**			
Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**		**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
HOST_EC Data/CMD Register	00h	00h	W	0	100h	0	R	
	04h	04h		1		0	R	
EC_HOST Data/AUX Data Register	00h	00h	R	0	100h	0	W	
				0	10Ch	0		
Keyboard Status Read Register	04h	04h	R	0	104h	3	R/W	
PCOBF Register	-	-	-	0	114h	0	R/W	
Keyboard Control Register	-	-	-	0	108h	0	R/W	
	Host Access			**N/A**	**EC Interface**			
Register Name	**Host Config. Index**	**SPB Offset**	**Host Type**		**EC Offset**	**Byte Lane**	**EC Type**	
Activate Register	30h	330h	R/W		330h	0	R/W	

Note 10-2 CMD is bit D3 in the Keyboard Status Read Register.

Note 10-3 All Registers listed in Table 10-3 are powered by VTR and reset by nSYS_RST.

10.7 8042 MSG Interface Configuration Registers

10.7.1 ACTIVATE REGISTER

TABLE 10-4: 8042 MSG Interface ACTIVATE REGISTER

HOST CONFIG INDEX	330h					8-bit	**HOST SIZE**	
EC OFFSET	NA					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R							R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	Reserved							Activate

ACTIVATE

0=1MHz Clock to the CPU_REST block is disabled, ONLY if the PGEN is not active

1=1MHz Clock to the CPU_REST block is enabled

10.8 8042 MSG Interface Runtime Registers

10.8.1 HOST_EC DATA / CMD REGISTER

This is an 8-bit HOST write-only register. When written with Data, the C/D status bit of the status register is cleared to zero and the IBF bit is set.

When written with a command, the C/D status bit of the status register is set to one and the IBF bit is set to a 1.

TABLE 10-5: HOST_EC DATA/CMD REGISTER

HOST I/O INDEX	00h Data 04h Command					8-bit	**HOST SIZE**	
EC OFFSET	100h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	W	W	W	W	W	W	W	W
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Data							

10.8.2 EC_HOST DATA / AUX DATA REGISTER

This is an 8-bit read-only register. When read, by the HOST, the PCOBF and/or AUXOBF interrupts are cleared and the OBF flag in the status register is cleared.

TABLE 10-6: EC_HOST DATA/AUX DATA REGISTER

HOST I/O INDEX	00h			8-bit			**HOST SIZE**	
EC OFFSET	100h Data or 10Ch Aux Data			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	Data							

10.8.3 KEYBOARD STATUS READ

This is an 8 bit read only register. Refer to the description of the Status Register (EC OFFSET 104h) for more information.

TABLE 10-7: KEYBOARD STATUS READ REGISTER

HOST I/O INDEX	4h			8-bit			**HOST SIZE**	
EC OFFSET	104h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R/W	R/W	R/W	R/W	R	R/W	R	R
BIT NAME	UD	UD	AUXOBF	UD	C/D	UD[1]	IBF	OBF

This register is read-only for the Host and read/write by the EC. The EC cannot write to bits 0, 1, or 3 of the Status register.

APPLICATION NOTE: the IBF and OBFbits are not cleared (‘0’) by hardware when VCC_PWRGD is asserted or when the LPC interface powers down; for example, following system changes state S3->S0, S5->S0, G3-> S0. To clear the IBF bit in firmware, read offset 100h; to clear the OBF bit in firmware, read offset 000h.

UD[1]

Read/Write by EC. These bits are user-definable. This bit is reset to ‘0’ when LRESET# pin signal function is asserted. See Table 2-7, “Host Interface,” on page 13.

UD

Read/Write by EC. These bits are user-definable.

C/D

Command Data - This bit specifies whether the input data register contains data or a command (“0” = data, “1” = command). During a host command write operation, this bit is set to “1”, during a host data write operation, this bit is set to “0”.

IBF

Input Buffer Full - This flag is set to “1” whenever the host system writes data or a command into the HOST_EC Data/CMD Registerr. Setting this flag activates the EC's EC_IBF interrupt if enabled. When the EC reads the HOST_EC Data / CMD Register, this bit is automatically reset and the interrupt is cleared.

OBF

The Output Buffer Full (OBF) bit is set when the EC writes a byte of Data or AUX Data into the EC_HOST Data / AUX Data Register. When the host reads the data, the OBF bit is automatically cleared by hardware and a EC_OBF interrupt is generated.

AUXOBF

Auxiliary Output Buffer Full - This flag is set to “1” whenever the EC writes AUX Data into EC_HOST Data / AUX Data Register. This flag is reset to “0” whenever the EC writes into the Data into EC_HOST Data / AUX Data Register.

TABLE 10-8: PCOBF REGISTER

HOST OFFSET	NA			8-bit			**HOST SIZE**	
EC OFFSET	114h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							R/W
BIT NAME	Reserved							PCOBF

Note 10-4 Refer to the PCOBF Description for information on this register. This is a “1” bit register (bits 1-7=0 on read).

10.9 EC-to-Host Keyboard Communication

The EC can write to the EC_HOST Data/AUX Data Register via EC OFFSET 100h or EC OFFSET 10Ch (Aux Host Data) respectively. A write to either of these addresses automatically sets bit 0 (OBF) in the Status register. A write to EC OFFSET 100h also sets PCOBF. A write to EC OFFSET 10Ch also sets AUXOBF. See Table 10-9.

TABLE 10-9: HOST-INTERFACE FLAGS

EC Address	Flag
EC OFFSET 100h (R/W)	PCOBF (KIRQ) output signal goes high
EC OFFSET 10Ch (W)	AUXOBF (MIRQ) output signal goes high

The HOST_EC Data / CMD Register and EC_HOST Data / AUX Data Register are each 8 bits wide. A write to this 8 bit register by the EC will load the EC_HOST Data / AUX Data Register, set the OBF flag and set the PCOBF output if enabled.

Note 10-5 Refer to the PCOBF and Keyboard Status Read Register descriptions for more information.

10.9.1 PCOBF DESCRIPTION

(The following description assumes that OBFEN = 1 in Keyboard Control Register on page 177); PCOBF is gated onto KIRQ. The KIRQ signal is a system interrupt which signifies that the EC has written to the KBD Data Read register via EC OFFSET100h. On power-up, PCOBF is reset to 0. PCOBF will normally reflect the status of writes to EC OFFSET 100h, if PCOBFEN (bit 2 of Configuration register “0”) = “0”. (KIRQ is normally selected as IRQ1 for keyboard support). PCOBF is cleared by hardware on a HOST read of the EC_HOST Data / AUX Data Register.

Additional flexibility has been added which allows firmware to directly control the PCOBF output signal, independent of data transfers to the host-interface data output register. This feature allows the MEC1609/MEC1609i to be operated via the host “polled” mode. This firmware control is active when PCOBFEN = 1 and firmware can then bring PCOBF high by writing a “1” to the LSB of the 1 bit data register, PCOBF, at EC OFFSET 114h. The firmware must also clear this bit by writing a “0” to the LSB of the 1 bit data register at EC OFFSET 114h.

The PCOBF register is also readable; bits 1-7 will return a “0” on the read back. The value read back on bit 0 of the register always reflects the present value of the PCOBF output. If PCOBFEN = 1, then this value reflects the output of the firmware latch at EC OFFSET 114h. If PCOBFEN = 0, then the value read back reflects the in-process status of write cycles to EC OFFSET 100h (i.e., if the value read back is high, the host interface output data register has just been written to). If OBFEN=0, then KIRQ is driven inactive (low).

10.9.2 AUXOBF1 DESCRIPTION

(The following description assumes that OBFEN = 1 in Keyboard Control Register on page 177); This bit is multiplexed onto MIRQ. The AUXOBF1/MIRQ signal is a system interrupt which signifies that the EC has written to the EC_HOST Data / AUX Data Register.

On power-up, after nSYS_RST, AUXOBF1 is reset to 0. AUXOBF1 will normally reflects the status of writes to EC OFFSET 10Ch. (MIRQ is normally selected as IRQ12 for mouse support). AUXOBF1 is cleared by hardware on a read of the Host Data Register. If OBFEN=0, then KIRQ is driven inactive (low).

TABLE 10-10: STATUS AND INTERRUPT BEHAVIOR OF WRITING TO OUTPUT DATA REGISTER

	Host I/F Status Register Bits			
Write to Register	AUXOBF (D5)	OBF (D0)	OBFEN=0	OBFEN=1
EC OFFSET 100h	0	1	KIRQ=0	KIRQ=1
EC OFFSET 10Ch	1	1	MIRQ=0	MIRQ=1

TABLE 10-11: OBFEN AND PCOBFEN EFFECTS ON KIRQ

OBFEN	PCOBFEN	
0	X	KIRQ is inactive and driven low
1	0	KIRQ = PCOBF@EC OFFSET 100h
1	1	KIRQ = PCOBF@EC OFFSET 114h

TABLE 10-12: OBFEN AND AUXH EFFECTS ON MIRQ

OBFEN	AUXH	
0	X	MIRQ is inactive and driven low
1	0	MIRQ = PCOBF@EC OFFSET 10Ch; Status Register D5 = User Defined
1	1	MIRQ = PCOBF@EC OFFSET 10Ch; Status Register D5 = Hardware Controlled

10.9.2.1 Keyboard Control

TABLE 10-13: KEYBOARD CONTROL REGISTER

HOST OFFSET	NA				8-bit		**HOST SIZE**	
EC OFFSET	108h				8-bit		**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AUXH	UD	OBFEN	UD	UD	PCOB-FEN	SAEN	UD

UD

User defined bit

AUXH

Aux in Hardware; when high, AUXOBF of the status register is set in hardware by a write to EC OFFSET 10Ch. When low, AUXOBF of the status register is a user defined bit (UD) and R/W.

OBFEN

When set,PCOBF is gated onto KIRQ and AUXOBF1 is gated onto MIRQ. When low, KIRQ and MIRQ are driven low. Software should not change this bit when OBF of the status register is equal to 1.

PCOBFEN

When high, PCOBF reflects whatever value was written to the PCOBF firmware latch assigned to 114h. When low, PCOBF reflects the status of writes to EC OFFSET 100h (the output data register).

SAEN

Software-assist enable. When set to “1,” SAEN allows control of the GATEA20 signal via firmware. If SAEN is reset to ‘0’, GATEA20 corresponds to either the last host-initiated control of GATEA20 or the firmware write to EC OFFSETs 108h or 10Ch.

10.10 Legacy Support

10.10.1 Port92-Legacy REGISTERS

The Table 10-14 is a register summary for the Port92-Legacy block. The LPC I/O address of the PORT92 Register is fixed. Each EC address is indicated as an SPB Offset from its AHB base address listed for the Port92-Legacy block in Table 10-2, “8042 Emulated Keyboard Controller Base Address Table,” on page 172. Each Configuration register access through the Host Access Port address via its LND indicated in Table 10-2 on page 172 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 10-14: Port92-Legacy SUPPORT REGISTER SUMMARY

	Host I/O Access			EC Interface			
Register Name	**Host I/O Address**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
GATEA20 Control Register	-	-	-	100h	0	R/W	
SETGA20L Register	-	-	-	108h	0	W	
RSTGA20L Register	-	-	-	10Ch	0	W	
PORT92 Register	92h	000h	R/W	000h	0	R/W	
	Host Access			**EC Interface**			
Register Name	**Host Config. Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	
PORT92 Enable Register	30h	3F0h	R/W	330h	0	R/W	

Note 10-6 CMD is bit D3 in the Keyboard Status Read Register.

All Registers listed in Table 10-14 are powered by VTR and reset by nSYS_RST except the PORT92 Register which is powered by VTR and reset by nSIO_RESET. (See Section , "iRESET OUT," on page 104.)

10.11 Configuration Registers

10.11.1 PORT 92 ENABLE

The MEC1609/MEC1609i supports LPC I/O writes to port 92h as a quick alternate mechanism for generating a CPU_RESET pulse or controlling the state of GATEA20.

TABLE 10-15: PORT92 ENABLE REGISTER

HOST CONFIG INDEX	30h				8-bit		**HOST SIZE**	
EC OFFSET	330h				8-bit		**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R							R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	Reserved							P92_EN

10.12 Runtime Registers

10.12.1 PORT 92

The MEC1609/MEC1609i supports LPC I/O writes to port HOST I/O address 92h as a quick alternate mechanism for generating a CPU_RESET pulse or controlling the state of GATEA20.

The PORT92 Register resides at HOST I/O address 92h and is used to support the alternate reset (ALT_RST#) and alternate GATEA20 (ALT_A20) functions. This register defaults to 00h on assertion of nSIO_RESET (See iRESET OUT on page 104).

Setting the Port92 Enable bit (PORT92 Enable Register) enables the Port92h Register. When Port92 is disabled, by clearing the Port92 Enable bit, then access to this register is completely disabled (I/O writes to host 92h are ignored and I/O reads float the system data bus SD[7:0]).

TABLE 10-16: PORT92 REGISTER

HOST I/O ADDRESS	0092h			8-bit			**HOST SIZE**	
EC OFFSET	00h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R						R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	Reserved						AL-T_GATEA20	AL-T_CPU_RESET

ALT_CPU_RESET

This bit provides an alternate means to generate a CPU_RESET pulse. The CPU_RESET output provides a means to reset the system CPU to effect a mode switch from Protected Virtual Address Mode to the Real Address Mode. This provides a faster means of reset than is provided through the EC keyboard controller. Writing a “1” to this bit will cause the ALT_RST# internal signal to pulse (active low) for a minimum of 6µs after a delay of 14µs. Before another ALT_RST# pulse can be generated, this bit must be written back to “0”.

ALT_GATEA20

This bit provides an alternate means for system control of the MEC1609/MEC1609i GATEA20 pin.

0=ALT_A20 is driven low

1=ALT_A20 is driven high

When Port 92 is enabled, writing a 0 to bit 1 of the PORT92 Register forces ALT_A20 low. ALT_A20 low drives GATEA20 low, if A20 from the keyboard controller is also low. When Port 92 is enabled, writing a 1 to bit 1 of the PORT92 Register forces ALT_A20 high. ALT_A20 high drives GATEA20 high regardless of the state of A20 from the keyboard controller.

10.12.2 GATE A20

The MEC1609/MEC1609i contains on-chip logic support for the GATEA20 hardware speed-up feature. GATEA20 is part of the control required to mask address line A20 to emulate 8086 addressing.

In addition to the ability for the host to control the GATEA20 output signal directly, a configuration bit called “SAEN” (Software Assist Enable, bit 1 of Keyboard Control Register is provided; when set, SAEN allows firmware to control the GATEA20 output.

When SAEN is set, a 1 bit register (GATEA20 Control Register) controls the GATEA20 output. The register bit allocation is shown in Table 10-13.

Note 10-7 Refer to the GATEA20 Control description for information on this register. This is a one bit register. (Bits 1-7=0 on read)

10.12.3 GATEA20 CONTROL

TABLE 10-17: GATEA20 CONTROL REGISTER

HOST OFFSET	NA			8-bit			**HOST SIZE**	
EC OFFSET	100h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							R/W
BIT NAME	Reserved							GATEA20

GATEA20

Writing a “0” into bid GATEA20 causes the GATEA20 output to go low, and vice versa.

Host control and firmware control of GATEA20 affect two separate register elements. Read back of GATEA20 through the use of EC OFFSET 100h reflects the present state of the GATEA20 output signal: if SAEN is set, the value read back corresponds to the last firmware-initiated control of GATEA20; if SAEN is reset, the value read back corresponds to the last host-initiated control of GATEA20.

Host control of the GATEA20 output is provided by the hardware interpretation of the “GATEA20 sequence” (see Table 10-18). The foregoing description assumes that the SAEN configuration bit is reset.

When the MEC1609/MEC1609i receives a “D1” command followed by data (via the host interface), the on-chip hardware copies the value of data bit 1 in the received data field to the GATEA20 host latch. At no time during this host-interface transaction will PCOBF or the IBF flag (bit 1) in the Keyboard Status Read Register be activated; for example, this host control of GATEA20 is transparent to firmware, with no consequent degradation of overall system performance. Table 10-18 details the possible GATEA20 sequences and the MEC1609/MEC1609i responses.

On VCC_POR, GATEA20 will be set.

An additional level of control flexibility is offered via a memory-mapped synchronous set and reset capability. Any data written to EC OFFSET 108h causes the GATEA20 host latch to be set; any data written to EC OFFSET 10Ch causes it to be reset. This control mechanism should be used with caution. It was added to augment the “normal” control flow as described above, not to replace it. Since the host and the firmware have asynchronous control capability of the host latch via this mechanism, a potential conflict could arise. Therefore, after using the EC OFFSET 108h and EC OFFSET 10Ch addresses, firmware should read back the GATEA20 status via EC OFFSET 100h (with SAEN = 0) to confirm the actual GATEA20 response.

TABLE 10-18: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES

Data Byte	R/W	D[0:7]	IBF Flag	GATEA20	Comments
1 0 1	W W W	D1 DF FF	0 0 0	Q 1 1	GATEA20 Turn-on Sequence
1 0 1	W W W	D1 DD FF	0 0 0	Q 0 0	GATEA20 Turn-off Sequence

TABLE 10-18: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES (CONTINUED)

Data Byte	R/W	D[0:7]	IBF Flag	GATEA20	Comments
1 1 0 1	W W W W	D1 D1 DF FF	0 0 0 0	Q Q 1 1	GATEA20 Turn-on Sequence(*)
1 1 0 1	W W W W	D1 D1 DD FF	0 0 0 0	Q Q 0 0	GATEA20 Turn-off Sequence(*)
1 1 1	W W W	D1 XX** FF	0 1 1	Q Q Q	Invalid Sequence

Note 10-8

- All examples assume that the SAEN configuration bit is 0.
- "Q" indicates the bit remains set at the previous state.
- *Not a standard sequence.
- **XX = Anything except D1.
- If multiple data bytes, set IBF and wait at state 0. Let the software know something unusual happened.
- For data bytes, only D[1] is used; all other bits are don't care.

TABLE 10-19: SETGA20L REGISTER

HOST OFFSET	NA			8-bit			**HOST SIZE**	
EC OFFSET	108h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	W							
BIT NAME								

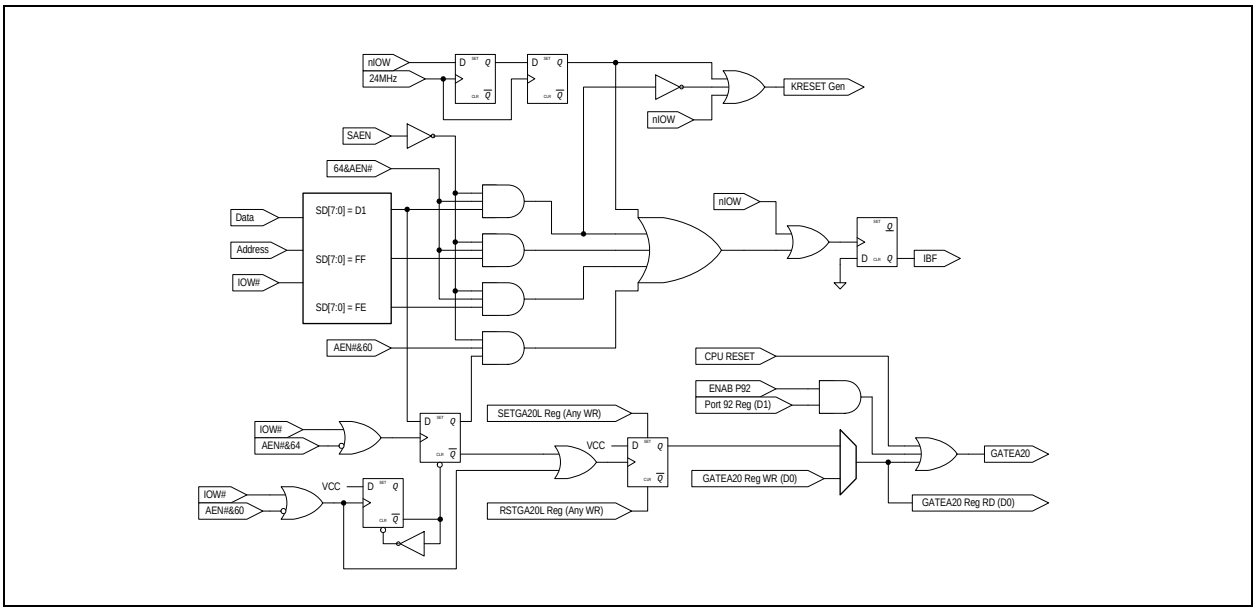
Note 10-9 Refer to the GATEA20 Hardware Speed-up description for information on this register. A write to this register sets GATEA20.

TABLE 10-20: RSTGA20L REGISTER

HOST OFFSET	NA					8-bit		**HOST SIZE**
EC OFFSET	10Ch					8-bit		**EC SIZE**
POWER	VTR					00h		nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	W							
BIT NAME								

Note 10-10 Refer to the GATEA20 Hardware Speed-up description for information on this register. A write to this register re-sets GATEA20.

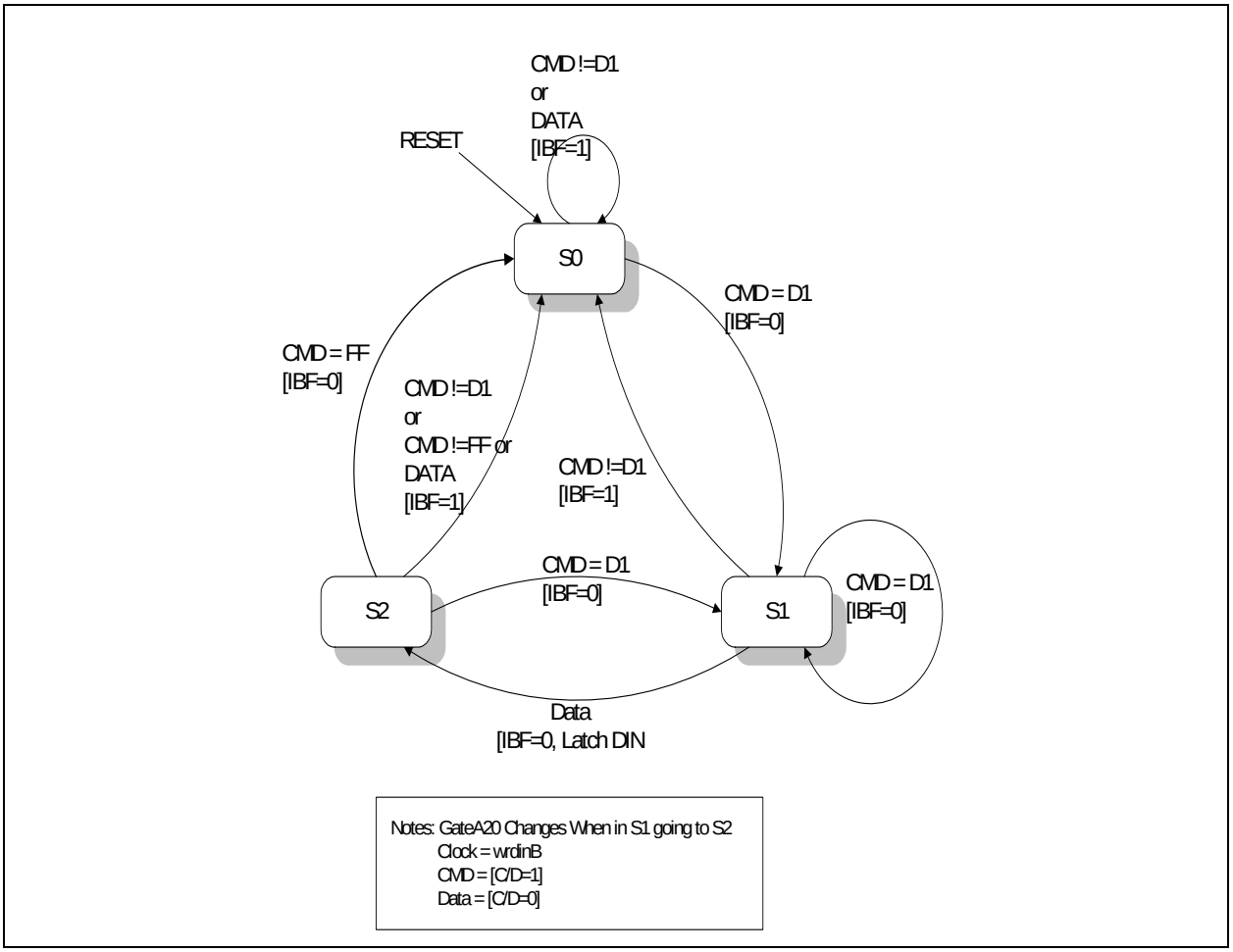
FIGURE 10-2: GATEA20 IMPLEMENTATION DIAGRAM



Note 10-11 Host Commands (FF, FE, & D1) do not cause IBF. The method of blocking IBF in Figure 10-2 is the nIOW not being asserted when FF, FE, & D1 Host commands are written".

The hardware GATEA20 state machine returns to state S1 from state S2 when CMD = D1 (Figure 10-3).

FIGURE 10-3: GATEA20 STATE MACHINE



10.13 CPU_RESET Hardware Speed-Up

The ALT_CPU_RESET bit generates, under program control, the ALT_RST# signal, which provides an alternate, means to drive the MEC1609/MEC1609i CPU_RESET pin which in turn is used to reset the Host CPU. The ALT_RST# signal is internally NANDed together with the KBDRESET# pulse from the KRESET Speed up logic to provide an alternate software means of resetting the host CPU.

Note 10-12 Before another ALT_RST# pulse can be generated, ALT_CPU_RESET must be cleared to “0” either by an nSIO_RESET (See iRESET OUT on page 104) or by a write to the PORT92 Register with bit 0 = “0”. A ALT_RST# pulse is not generated in the event that the ALT_CPU_RESET bit is cleared and set before the prior ALT_RESET# pulse has completed.

Note 10-13 this function is qualified by the SLP_EN signal or the Activate bit. If either of these signals goes to ‘0’, then the 1MHz clocks source is disabled if the PGEN is not currently active. See EC Blocks Sleep Enables/Clock Required Registers Bit Names on page 107 & Section 5.4.7.5.4, "Block Sleep Enables," on page 89.

FIGURE 10-4: CPU_RESET IMPLEMENTATION DIAGRAM

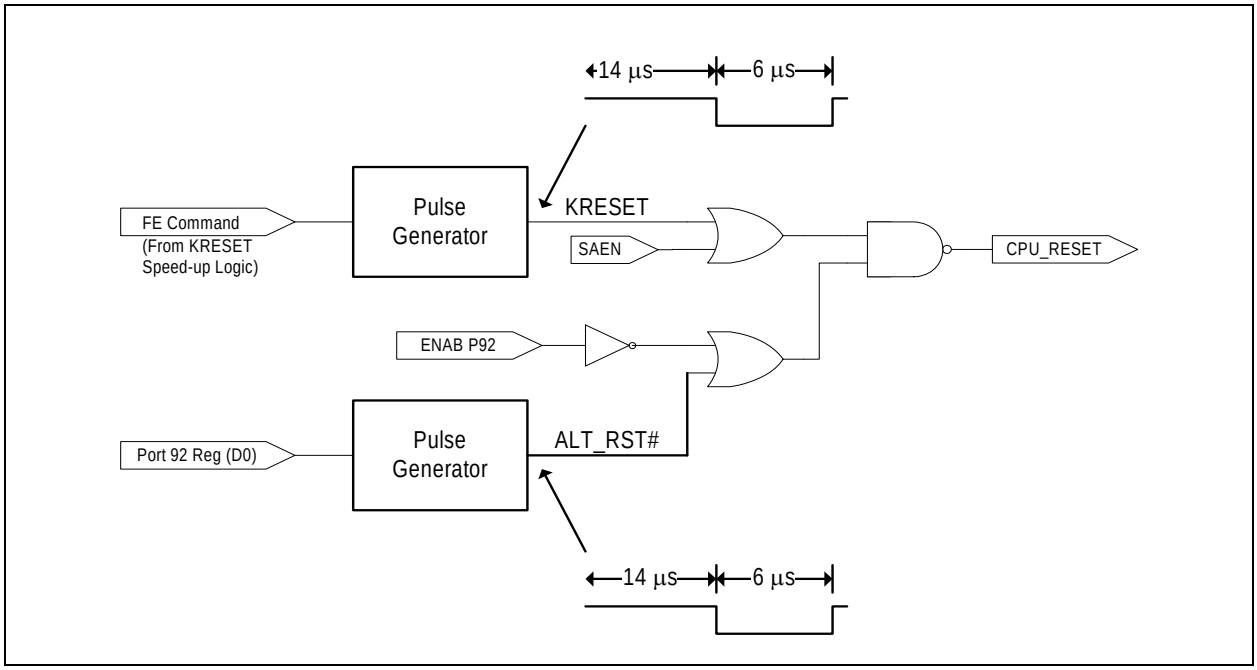


FIGURE 10-5: CPU_RESET TIMING

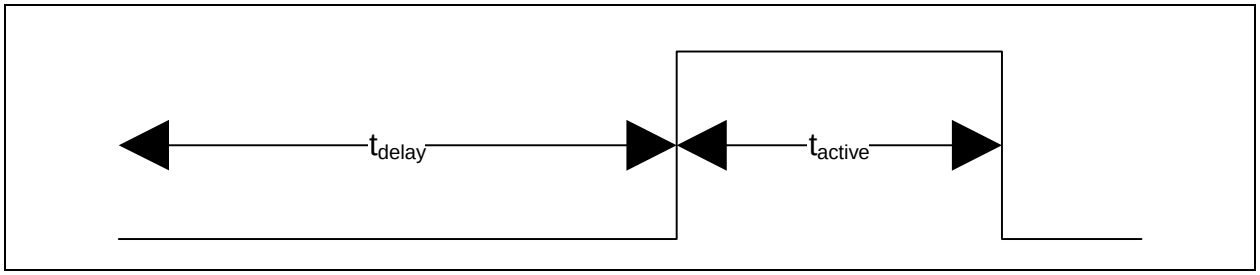


TABLE 10-21: CPU_RESET TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{delay}	Delay prior to active pulse	14	15	15.5	µs
t_{active}	Active pulse width	6	8	8.5	µs

Note 10-14 Figure 10-5 & Table 10-21 refers to Figure 10-4 in which CPU_RESET is the inverse of ALT_RST# & KRESET.

Note 10-15 The KBRST pin function is the output of CPU_RESET described in Section 10.13, "CPU_RESET Hardware Speed-Up," on page 183.

11.0 ACPI PM1 BLOCK INTERFACE

11.1 General Description

The MEC1609/MEC1609i supports ACPI as described in this section. These features comply with the ACPI Specification, Revision 1.0, through a combination of hardware and EC software.

The MEC1609/MEC1609i implements the ACPI fixed registers but includes only those bits that apply to the power button sleep button and RTC alarm events. The ACPI WAK_ SLP_TYPx, and SLP_ bits are also supported.

11.2 Power, Clocks and Reset

11.2.1 POWER DOMAIN

This block is powered by VTR

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

11.2.2 CLOCKS

This block has one clock input, the LPC Bus Clock.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

11.2.3 POWER ON RESET

This block is reset on a nSYS_RST. After nSYS_RST is asserted, all registers are set to ‘0’.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

11.3 Interrupts

11.3.1 SCI INTERRUPTS TO THE HOST

The functions described in the following sub-sections can generate a SCI event on the EC_SCI# pin. In the MEC1609/MEC1609i, an SCI event is considered the same as an ACPI wakeup or runtime event. The EC can also generate a SCI on the EC_SCI# pin by setting the EC_SCI_ bit in the EC_SCI# Pin Interface on page 192.

11.3.2 INTERRUPTS TO THE EC

An Interrupt is generated to the EC on PM1_CTL2 bit of GIRQ13 Source Register by the Host writing to Power Management 1 Control Register 2 (PM1_CNTRL 2).

An Interrupt is generated to the EC on PM1_EN2 bit of GIRQ13 Source Register by the Host writing to Power Management 1 Enable Register 2 (PM1_EN 2).

An Interrupt is generated to the EC on PM1_STS2 bit of GIRQ13 Source Register by the Host writing to Power Management 1 Status Register 2 (PM1_STS 2).

11.3.3 ACPI PM1 BLOCK SCI EVENT-GENERATING FUNCTIONS

11.3.3.1 Power Button with Override

The power button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. It also has a status and enable bit in the PM1_BLK of registers to indicate and control the power button override (fail-safe) event. These bits are not required by ACPI. The power button override event status bit is software Read/Writable by the EC; the enable bit is software read-only by the EC. The enable bit for the override event is located at bit 1 in the Power Management 1 Control Register 2 (PM1_CNTRL 2).

The PWRBTN_ bit is set by the Host to enable the generation of an SCI due to the power button event. The status bit is set by the EC when it generates a power button event and is cleared by the Host writing a ‘1’ to this bit (writing a ‘0’ has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

11.3.3.2 Sleep Button

The sleep button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC.

The SLPBTN_ bit is set by the Host to enable the generation of an SCI due to the sleep button event. The status bit is set by the EC when it generates a sleep button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

11.3.4 RTC ALARM

The ACPI specification requires that the RTC alarm generate a hardware wake-up event from the sleeping state. The RTC alarm can be enabled as an SCI event and its status can be determined through bits in the PM1_BLK of registers. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC.

The RTC_ bit is set by the Host to enable the generation of an SCI due to the RTC alarm event. The status bit is set by the EC when the RTC generates an alarm event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

11.4 Registers

Each instance of the ACPI PM1 Block Interface has its own Logical Device Number, and Base Address as indicated in Table 11-1.

TABLE 11-1: ACPI PM1 Block Interface BASE ADDRESS TABLE

ACPI PM1 Block Interface **Instance**	Table 3-1 on page 48 **LDN**	**AHB Base Address**
ACPI PM1 Block Interface	6h	FF_1800h

The ACPI register model consists of a number of fixed register blocks that perform designated functions. A register block consists of a number of registers that perform Status, Enable and Control functions. The ACPI specification deals with events (which have an associated interrupt status and enable bits, and sometimes an associated control function) and control features. The status registers illustrate what defined function is requesting ACPI interrupt services (SCI). Any status bit in the ACPI specification has the following attributes:

Status bits are only set through some defined hardware or EC event.

Unless otherwise noted, status bits are cleared by the system writing a "1" to that bit position, and upon nSYS_RST. Writing a '0' has no effect.

Status bits only generate interrupts while their associated bit in the enable register is set.

Function bit positions in the status register have the same bit position in the enable register (there are exceptions to this rule, special status bits have no enables).

Note that this implies that if the respective enable bit is reset and the hardware event occurs, the respective status bit is set; however no interrupt is generated until the enable bit is set. This allows software to test the state of the event (by examining the status bit) without necessarily generating an interrupt. There are a special class of status bits that have no respective enable bit, these are called out specifically, and the respective enable bit in the enable register is marked as reserved for these special cases.

The enable registers allow the setting of the status bit to generate an interrupt (under EC control). As a general rule, there is an enable bit in the enable register for every status bit in the status register. The control register provides special controls for the associated event, or special control features that are not associated with an interrupt event. The order of a register block is the status registers, followed by enable registers, followed by control registers.

The registers in the MEC1609/MEC1609i ACPI PM1 Block Interface occupy eight addresses in the host I/O space and are specified as offsets from the ACPI PM1 Block base address (Table 11-1).

TABLE 11-2: ACPI PM1 Block Interface REGISTER SUMMARY

	Host Access			EC Access			
Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Power Management 1 Status Register 1 (PM1_STS) 1	0h	00h	R	100h	0	R	
Power Management 1 Status Register 2 (PM1_STS 2)	1h	01h	R/WC	101h	1	R/W	Table 11-4 Note 11-4
Power Management 1 Enable Register 1 (PM1_EN 1)	2h	02h	R	102h	2	R	
Power Management 1 Enable Register 2 (PM1_EN 2)	3h	03h	R/W	103h	3	R	Table 11-6 Note 11-5
Power Management 1 Control Register 1 (PM1_CNTRL 1)	4h	04h	R	104h	0	R	
Power Management 1 Control Register 2 (PM1_CNTRL 2)	5h	05h	R/W	105h	1	R	Table 11-8 Note 11-6
Power Management 2 Control Register 1 (PM2_CNTRL 1)	6h	06h	R	106h	2	R	Note 11-2
Power Management 2 Control Register 2 (PM2_CNTRL 2)	7h	07h	R	107h	3	R	Note 11-2
EC_PM_STS Register	-	-	-	110h	0	R/W	

Note 11-1 Byte 0 of this register is reserved.

Note 11-2 These registers return ‘0’ when read, writes have no effect.

11.4.1 POWER MANAGEMENT 1 STATUS 1 (PM1_STS 1)

TABLE 11-3: POWER MANAGEMENT 1 STATUS REGISTER 1 (PM1_STS) 1

HOST OFFSET	0h			8-bit			**HOST SIZE**	
EC OFFSET	100h						**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.4.2 POWER MANAGEMENT 1 STATUS 2 (PM1_STS 2)

TABLE 11-4: POWER MANAGEMENT 1 STATUS REGISTER 2 (PM1_STS 2)

HOST OFFSET	1h				8-bit		**HOST SIZE**	
EC OFFSET	101h						**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R	R	R	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/W	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	WAK_STS	Reserved			PWRBTNOR_STS	RTC_STS	SLPBTN_STS	PWRBTN_STS

Note 11-3 These bits are set/cleared by the EC directly i.e., writing ‘1’ sets the bit and writing ‘0’ clears it. These bits can also be cleared by the Host software writing a one to this bit position and by nSYS_RST. Writing a 0 by the Host has no effect.

Note 11-4 An interrupt (PM1_CTL2) is generated to the EC when the Host writes to this register.

PWRBTN_STS

This bit can be set or cleared by the EC to simulate a Power button status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.

SLPBTN_STS

This bit can be set or cleared by the EC to simulate a Sleep button status if the sleep state is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.

RTC_STS

This bit can be set or cleared by the EC to simulate a RTC status. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.

PWRBTNOR_STS

This bit can be set or cleared by the EC to simulate a Power button override event status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated hardware event under software control.

WAK_STS

This bit can be set or cleared by the EC. The Host writing a one to this bit can also clear this bit.

11.4.3 POWER MANAGEMENT 1 ENABLE 1 (PM1_EN 1)

TABLE 11-5: POWER MANAGEMENT 1 ENABLE REGISTER 1 (PM1_EN 1)

HOST OFFSET	02h			8-bit			**HOST SIZE**	
EC OFFSET	102h						**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.4.4 POWER MANAGEMENT 1 ENABLE 2 (PM1_EN 2)

TABLE 11-6: POWER MANAGEMENT 1 ENABLE REGISTER 2 (PM1_EN 2)

HOST OFFSET	03h			8-bit			**HOST SIZE**	
EC OFFSET	103h						**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R/W	R/W	R/W
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved					RTC_EN	SLPBTN_EN	PWRBTN_EN

Note 11-5 An interrupt (PM1_EN2) is generated to the EC when the Host writes to this register.

PWRBTN_EN

This bit can be read or written by the Host. It can be read by the EC.

SLPBTN_EN

This bit can be read or written by the Host. It can be read by the EC.

RTC_EN

This bit can be read or written by the Host. It can be read by the EC.

11.4.5 POWER MANAGEMENT 1 CONTROL 1 (PM1_CNTRL 1)

TABLE 11-7: POWER MANAGEMENT 1 CONTROL REGISTER 1 (PM1_CNTRL 1)

HOST OFFSET	04h			8-bit				**HOST SIZE**
EC OFFSET	104h							**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return '0' when read.

11.4.6 POWER MANAGEMENT 1 CONTROL 2 (PM1_CNTRL 2)

TABLE 11-8: POWER MANAGEMENT 1 CONTROL REGISTER 2 (PM1_CNTRL 2)

HOST OFFSET	5h			8-bit			**HOST SIZE**	
EC OFFSET	105h						**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	W	R/W			R/W	R/W
EC TYPE	R	R	R/WC	R			R	R
BIT NAME	Reserved		SLP_ EN	SLP_TYPx			PWRBTNOR_ EN	Reserved

Note 11-6 An interrupt (PM1_CTL2) is generated to the EC when the Host writes to this register.

PWRBTNOR_EN

This bit can be set or cleared by the Host, read by the EC.

SLP_TYPX

These bits can be set or cleared by the Host, read by the EC.

SLP_EN

Refer to Table 11-9, "SLP_ Definition".

TABLE 11-9: SLP_ DEFINITION

Host / EC	R/W	Description
Host	Read	Always reads 0
	Write	Writing a 0 has no effect, Writing a 1 sets this bit
EC	Read	Reads the value of the bit
	Write	Writing a 0 has no effect, Writing a 1 clears this bit

11.4.7 POWER MANAGEMENT 2 CONTROL 1 (PM2_CNTRL 1)

TABLE 11-10: POWER MANAGEMENT 2 CONTROL REGISTER 1 (PM2_CNTRL 1)

HOST OFFSET	06h					8-bit	**HOST SIZE**	
EC OFFSET	106h						**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.4.8 POWER MANAGEMENT 2 CONTROL 2 (PM2_CNTRL 2)

TABLE 11-11: POWER MANAGEMENT 2 CONTROL REGISTER 2 (PM2_CNTRL 2)

HOST OFFSET	07h					8-bit	**HOST SIZE**	
EC OFFSET	107h						**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.5 EC_SCI# Pin Interface

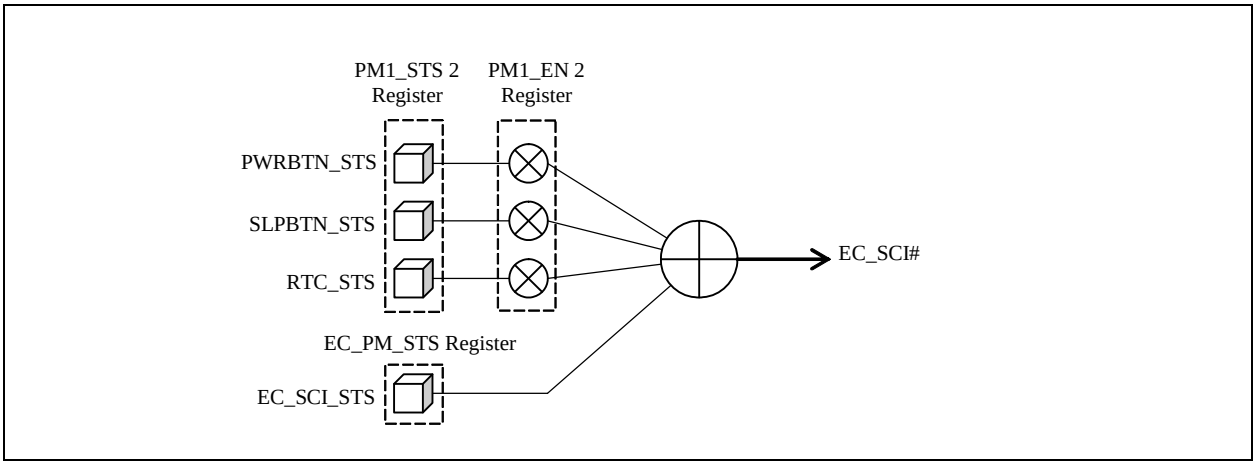
The EC_SCI# pin logic hardware is shown below in Figure 11-1.

Any or all of the PWRBTN, SLPBTN_STS, and RTC_ bits in the Power Management 1 Status Register 2 (PM1_STS 2) can assert the EC_SCI# pin if enabled by the PWRBTN_, SLPBTN_, and RTC_ bits in the PM1_EN 2 register.

The EC_SCI_ bit can assert the EC_SCI# pin at any time, without being enabled. The EC_SCI_ bit is located in the EC_PM_STS Register.

The EC_SCI_ bit is in the MEC1609/MEC1609i and is read/write by the EC. If the EC_SCI_ bit is “1”, an interrupt is generated on the EC_SCI# pin.

FIGURE 11-1: HARDWARE EC_SCI# INTERFACE



11.5.1 EC_PM_STS REGISTER

TABLE 11-12: EC_PM_STS REGISTER

HOST OFFSET							**HOST SIZE**	
EC OFFSET	110h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	UD[6:0							EC_SCI_ STS

EC_SCI_STS

If the EC_SCI_ bit is “1”, an interrupt is generated on the EC_SCI# pin.

UD[6:0]

User-defined bits. This bits do not generate an interrupt.

12.0 MAILBOX REGISTER INTERFACE

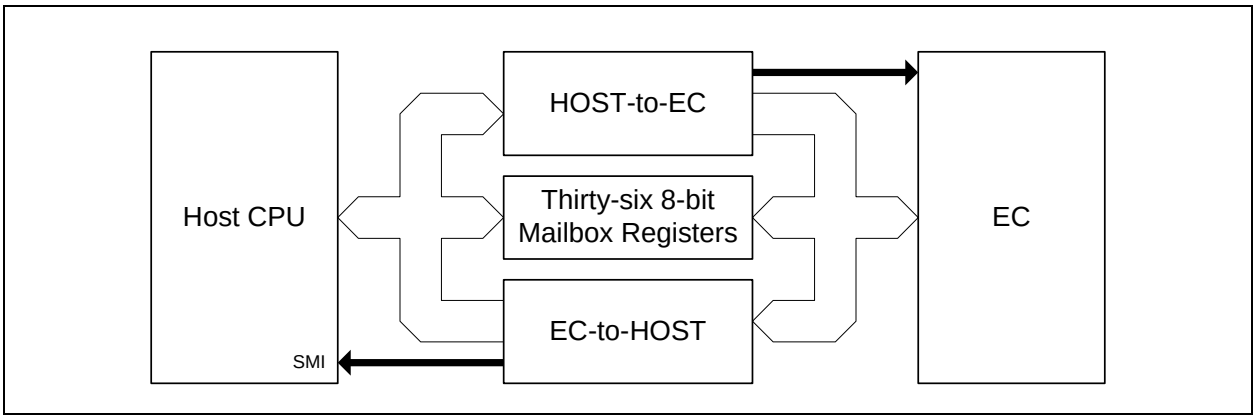
12.1 General Description

The MailBox Register Interface provides a standard run-time mechanism for the host to communicate with the Embedded Controller (EC) and other logical components in the MEC1609/MEC1609i (Figure 12-1). The Mailbox Registers Interface includes a total of 36 index-addressable 8-bit registers and a Mailbox Registers Interface Host Access Port. Thirty-two of 36 index-addressable 8-bit registers are EC Mailbox registers. The Mailbox Registers Interface Host Access Port consists of two 8-bit run-time registers that occupy two addresses in the HOST I/O space. The Mailbox Registers Interface Host Access Port is used by the host to access the 36 index-addressable 8-bit registers.

Note:	In this specification, host access to registers in the Mailbox Registers Interface through the host access port are identified by the prefix MBX in front of a hexadecimal index address.

12.1.1 BLOCK DIAGRAM

FIGURE 12-1: MAILBOX BLOCK DIAGRAM



12.2 Power, Clocks and Reset

12.2.1 POWER DOMAIN

This block is powered by the VTR power supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

12.2.2 CLOCKS

This block has one clock input, the LPC Bus Clock.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

12.2.3 RESET

This block is reset when nSYS_RST is asserted.

In addition the MBX_INDEX Register & MBX_DATA Register are reset when VCC_PWRGD Signal Pin function is de-asserted.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

12.3 Interrupts

The MailBox Register Interface can generate an interrupt event for the HOST-to-EC events. See HOST-to-EC Mailbox Register on page 198. The interrupt source is routed onto the MBX bit in the GIRQ15 Source Register and is a level, active high signal.

12.3.1 MAILBOX REGISTER INTERFACE (LDN 0H) SIRQ ROUTING

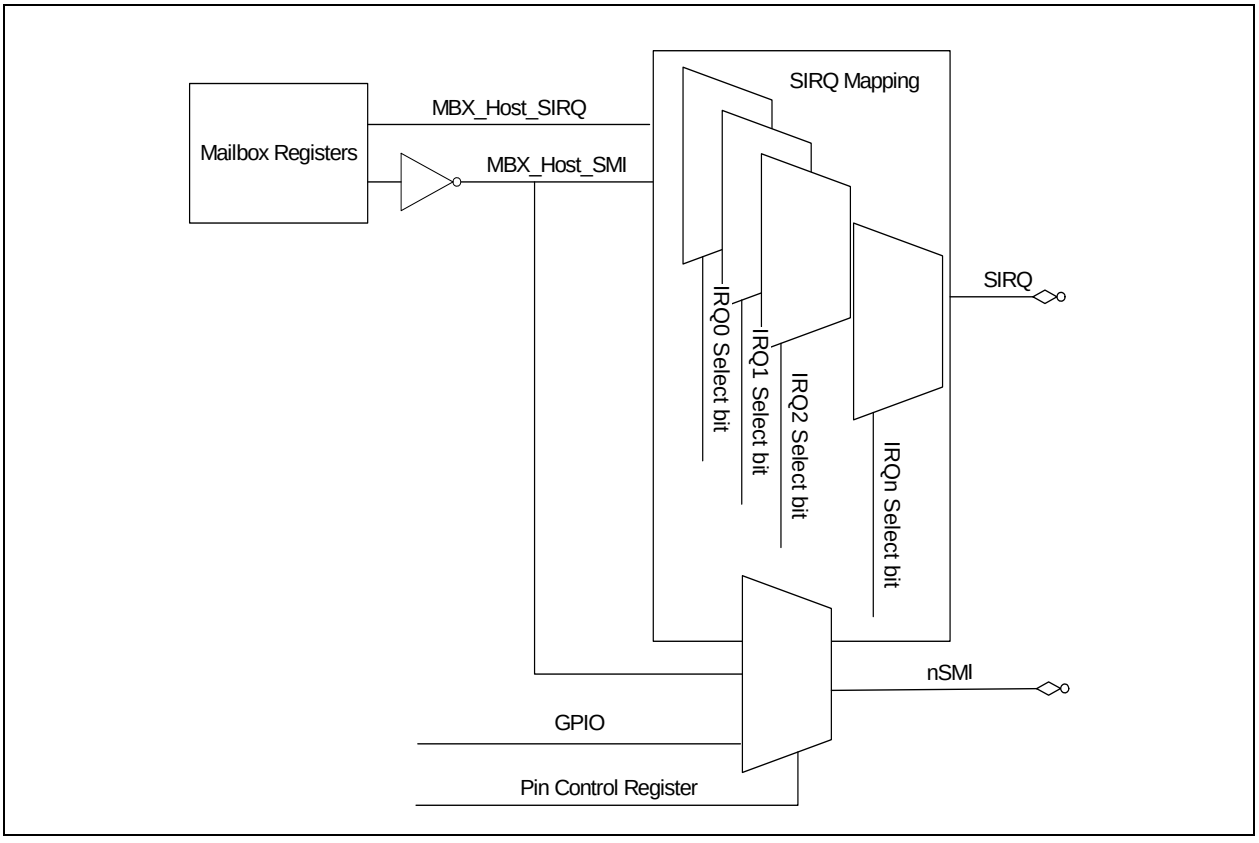
The MailBox Register Interface can generate a SIRQ event for the EC-to-HOST EC events. See HOST-to-EC Mailbox Register on page 198. This interrupt is routed to the SIRQ block (see Section 4.8.2, "SERIRQ Configuration Registers," on page 63). For this interrupt, the SELECT on page 64 is cleared to '0' in the Interrupt Configuration Register for the selected SIRQ frame.

The MailBox Register Interface can generate a SMI event from the SMI Interrupt Source Register on page 199. The SMI event can be routed to any frame in the SIRQ stream and to the nSMI pin. To enable SMI routing to the SIRQ stream, the bit SELECT on page 64 is set to '1' in the Interrupt Configuration Register for the selected SIRQ frame. The SMI event can be routed to nSMI pin by selecting the nSMI signal function in the associated Pin Control Register on page 337.

The SMI event produces a standard active low on the serial IRQ stream and active low on the open drain nSMI pin. See FIGURE 12-2: Mailbox SIRQ and SMI routing on page 194.

See Section 4.8.2, "SERIRQ Configuration Registers," on page 63.

FIGURE 12-2: MAILBOX SIRQ AND SMI ROUTING



12.4 Registers Summary

The MailBox Register Interface has its own Logical Device Number, and Base Address as indicated in Table 12-1. The Host LPC I/O addresses for the MailBox Register Interface are selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through Host Access Configuration Port (see Section 4.5.1, "Host Access Port," on page 55.)

Table 12-2 is a register summary for the MailBox Register Interface block.

TABLE 12-1: MailBox Register Interface BASE ADDRESS TABLE

MailBox Register Interface Instance	LDN from (Table 3-2 on page 48)	AHB Base Address
Mailbox Interface	0h	FF_0000h

Note: The Host LPC I/O addresses for this instance is selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through the Host Access Configuration Port. (See Section 4.5.1, "Host Access Port," on page 55.)

The Table 12-2 is a register summary for one instance of the MailBox Register Interface. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register is accessed through the Host Access Port is via its LDN indicated in Table 12-1 on page 195 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 12-2: MailBox Register Interface REGISTER SUMMARY

		Host I/O Access			EC Interface			
	Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
	MBX_INDEX Register	00h	00h	R/W	-	-	-	
	MBX_DATA Register	01h	01h	R/W	-	-	-	
		MAILBOX INDEX						
1.	HOST-to-EC Mailbox Register	MBX 00h	-	R/W	100h	0	R/WC	Note 12-1
2.	EC-to-Host Mailbox Register	MBX 01h	-	R/WC	104h	0	R/W	Note 12-2
3.	SMI Interrupt Source Register	MBX 02h	-	Table 12-7	108h	0	Table 12-7	
4.	SMI Interrupt Mask Register	MBX 03H	-	R/W	10Ch	0	R/W	
5.	**Mailbox register [0]**	MBX10h	-	R/W	110h	0	R/W	
6.	**Mailbox register [1]**	MBX11h	-			1		
7.	**Mailbox register [2]**	MBX12h	-			2		
8.	**Mailbox register [3]**	MBX13h	-			3		
9.	**Mailbox register [4]**	MBX14h	-	R/W	114h	0	R/W	
10.	**Mailbox register [5]**	MBX15h	-			1		
11.	**Mailbox register [6]**	MBX16h	-			2		
12.	**Mailbox register [7]**	MBX17h	-			3		

TABLE 12-2: MailBox Register Interface REGISTER SUMMARY (CONTINUED)

		Host I/O Access			EC Interface			
	Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
13.	**Mailbox register [8]**	MBX18h	-	R/W	118h	0	R/W	
14.	**Mailbox register [9]**	MBX19h	-			1		
15.	**Mailbox register (A]**	MBX1Ah	-			2		
16.	**Mailbox register [B]**	MBX1Bh	-			3		
17.	**Mailbox register [C]**	MBX1Ch	-	R/W	11Ch	0	R/W	
18.	**Mailbox register [D]**	MBX1Dh	-			1		
19.	**Mailbox register [E]**	MBX1Eh	-			2		
20.	**Mailbox register [F]**	MBX1Fh	-			3		
21.	**Mailbox register [10]**	MBX20h	-	R/W	120h	0	R/W	
22.	**Mailbox register [11])**	MBX21h	-			1		
23.	**Mailbox register [12]**	MBX22h	-			2		
24.	**Mailbox register [13]**	MBX23h	-			3		
25.	**Mailbox register [14]**	MBX24h	-	R/W	124h	0	R/W	
26.	**Mailbox register [15]**	MBX25h	-			1		
27.	**Mailbox register [16]**	MBX26h	-			2		
28.	**Mailbox register [17]**	MBX27h	-			3		
29.	**Mailbox register [18]**	MBX28h	-	R/W	128h	0	R/W	
30.	**Mailbox register [19]**	MBX29h	-			1		
31.	**Mailbox register [1A]**	MBX2Ah	-			2		
32.	**Mailbox register [1B]**	MBX2Bh	-			3		
33.	**Mailbox register [1C]**	MBX2Ch	-	R/W	12Ch	0	R/W	
34.	**Mailbox register [1D]**	MBX2Dh	-			1		
35.	**Mailbox register [1E]**	MBX2Eh	-			2		
36.	**Mailbox register [1F]**	MBX2Fh	-			3		

Note 12-1 Interrupt is cleared when read by the EC.

Note 12-2 Interrupt is cleared when read by the host.

12.4.1 MAILBOX REGISTERS INTERFACE HOST ACCESS PORT

The Mailbox registers access port is two runtime registers that occupy two addresses in the Host I/O space: MBX_INDEX Register & MBX_DATA Register.

To access a Mailbox register once the Mailbox Registers Interface Base Address has been initialized, write the Mailbox register index address to the MBX Index port and read or write the Mailbox register data from the MBX data port.

See Table 12-2, “MailBox Register Interface Register Summary,” on page 195.

12.4.2 MAILBOX CONTROL REGISTERS

Mailbox Register, HOST-to-EC, and Mailbox Register, EC-to-HOST, are specifically designed to pass commands between the host and the EC (FIGURE 12-1: on page 193). If enabled, these registers can generate interrupts.

Mailbox Register and Mailbox Register are not dual-ported, so the HOST BIOS and Keyboard BIOS must be designed to properly share these registers. When the host performs a write of the HOST-to-EC mailbox register, an interrupt will be generated and seen by the EC if unmasked. When the EC writes FF to the HOST-to-EC mailbox register, resets the register to 00h, providing a simple means for the EC to inform the host that an operation has been completed.

When the EC writes the EC-to-HOST mailbox register, an SMI may be generated and seen by the host if unmasked. When the Host CPU writes FFh to the EC-to-HOST mailbox register, the EC-to-HOST register resets to 00h, providing a simple means for the host to inform that EC that an operation has been completed.

PROGRAMMER'S NOTE: The protocol used to pass commands back and forth through the Mailbox Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Mailbox registers to gain access to all of the EC registers.

12.5 Register Details

12.5.1 MAILBOX INDEX REGISTER

TABLE 12-3: MBX_INDEX REGISTER

HOST OFFSET	00h			8-Bit		**HOST SIZE**		
EC OFFSET	NA					**EC SIZE**		
POWER	VTR			00h 00h		nSYS_RST **DEFAULT** **&** VCC_PWRGD **DE-ASSERTION**		
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	INDEX[7:0]							

12.5.2 MAILBOX DATA REGISTER

TABLE 12-4: MBX_DATA REGISTER

HOST OFFSET	01			8-Bit		**HOST SIZE**		
EC OFFSET	NA					**EC SIZE**		
POWER	VTR			00h 00h		nSYS_RST **DEFAULT** **&** VCC_PWRGD **DE-ASSERTION**		
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	DATA[7:0]							

12.5.3 HOST-TO-EC MAILBOX REGISTER

TABLE 12-5: HOST-TO-EC MAILBOX REGISTER

HOST OFFSET	MBX_00h			8-Bit				**HOST SIZE**
EC OFFSET	100h			8-Bit				**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	HOST_EC_MBOX[7:0]							

HOST_EC_MBOX[7:0]

If enabled, an interrupt to the EC marked by the MBX bit in the GIRQ15 Source Register will be generated whenever the Host writes this register.

This register is cleared when written with FFh.

12.5.4 EC-TO-HOST MAILBOX REGISTER

TABLE 12-6: EC-TO-HOST MAILBOX REGISTER

HOST OFFSET	MBX_01h			8-Bit				**HOST SIZE**
EC OFFSET	104h			8-Bit				**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_HOST_MBOX[7:0]							

EC_HOST_MBOX[7:0]

An EC write to this register will set bit EC_WR in the SMI Interrupt Source Register to ‘1b’. If enabled, setting bit EC_WR to ‘1b’ generates a Host SMI.

This register is cleared when written with FFh.

12.5.5 SMI INTERRUPT SOURCE REGISTER

TABLE 12-7: SMI INTERRUPT SOURCE REGISTER

HOST OFFSET	MBX_02h			8-Bit			**HOST SIZE**	
EC OFFSET	108h			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
BIT NAME	EC_SWI[6:0]							EC_WR

EC_WR

This bit is set autonomously when the EC-to-Host Mailbox Register has been written. An SMI to the Host is generated when any bit in this register (EC_WR or any bit in EC_SWI[6:0]) is ‘1b’ and the corresponding bit in the SMI Interrupt Mask Register register is ‘1b’.

This bit is automatically cleared by a read of the EC-to-Host Mailbox Register. The bit is also cleared when written with a ‘1b’, by either the Host or the EC.

EC_SWI[6:0]

The EC can generate an SMI to the Host by writing any non-zero value to this field.

Each bit in this field is cleared when written with a ‘1b’.

12.5.6 SMI INTERRUPT MASK REGISTER

TABLE 12-8: SMI INTERRUPT MASK REGISTER

HOST OFFSET	MBX_03h			8-Bit			**HOST SIZE**	
EC OFFSET	10Ch			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_SWI_EN[6:0]							EC_WR_EN

EC_WR_EN

If this bit is ‘1b’, bit EC_WR in the SMI Interrupt Source Register is enabled.

EC_SWI_EN[6:0]

Each bit that is set to ‘1b’ in this field enables the corresponding bit in the EC_SWI[6:0] field n the SMI Interrupt Source Register.

13.0 TWO PIN SERIAL PORT (UART)

13.1 General Description

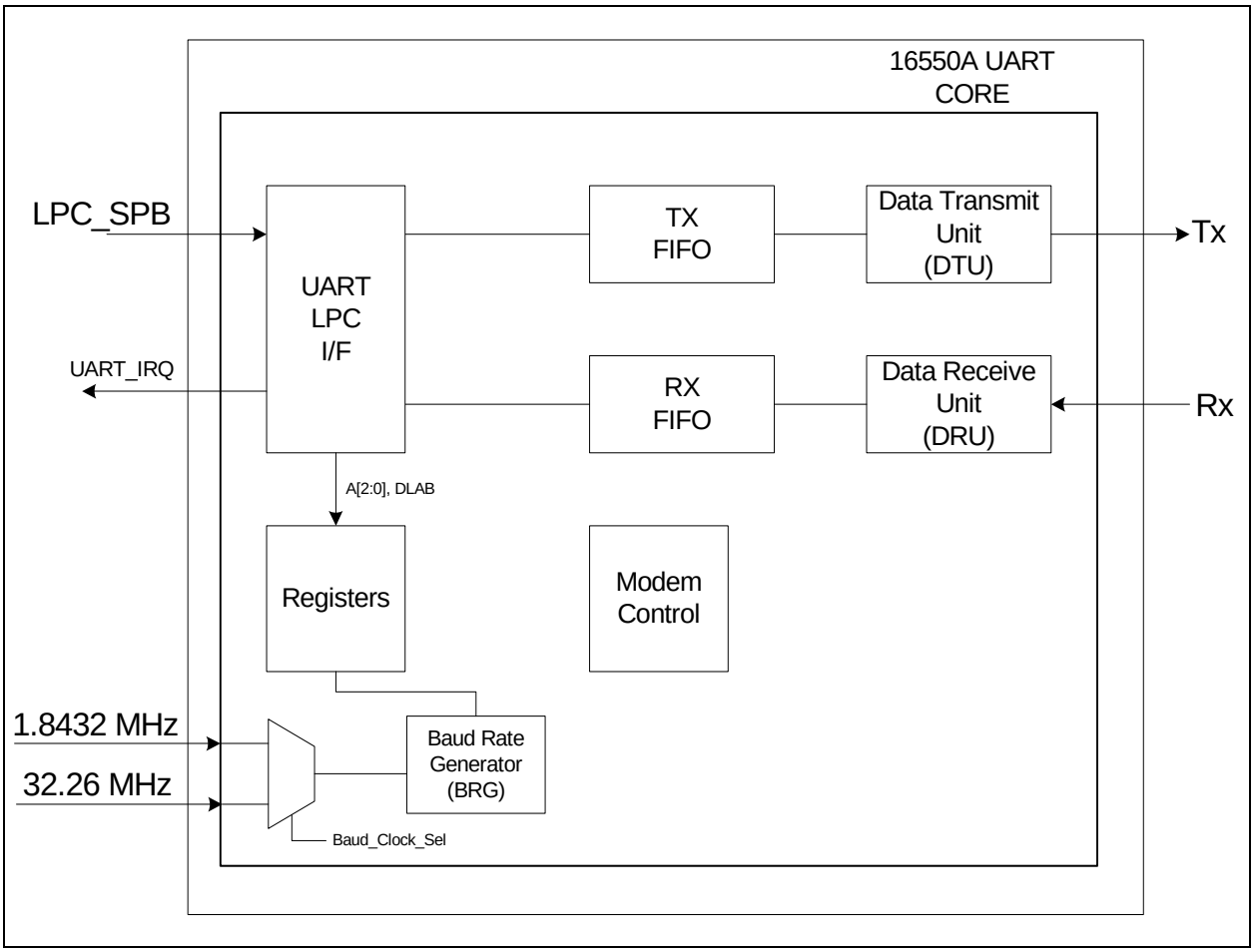
The MEC1609/MEC1609i incorporates one full function UART. The UART is compatible with the 16450, the 16450 ACE registers and the 16C550A. The UART performs serial-to-parallel conversion on received characters and parallel-to-serial conversion on transmit characters. Two sets of baud rates are provided. When the 1.8432 MHz source clock is selected, standard baud rates from 50 to 115.2K are available. When the source clock is 32.26 MHz, baud rates from 126K to 2,016K are available. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. The UART contains a programmable baud rate generator that is capable of dividing the input clock or crystal by a number from 1 to 65535. The UART is also capable of supporting the MIDI data rate. Refer to the Configuration Registers for information on disabling, powerdown and changing the base address of the UART. The interrupt from a UART is enabled by programming OUT2 of the UART to a logic “1”. OUT2 being a logic “0” disables that UART's interrupt. The UART is accessible by both the Host and the EC.

13.1.1 FEATURES

- Programmable word length, stop bits and parity
- Programmable baud rate generator
- Interrupt generator
- Loop-back mode
- Interface registers
- 16-byte Transmit FIFO
- 16-byte Receive FIFO
- Multiple clock sources
- VTR & VCC operation
- Pin Polarity control
- Low power sleep mode

13.1.2 BLOCK DIAGRAM

FIGURE 13-1: SERIAL PORT (UART) BLOCK DIAGRAM



13.1.3 BLOCK DIAGRAM SIGNAL LIST

TABLE 13-1: SERIAL PORT (UART) REGISTER INTERFACE PORT LIST

Signal Name	Direction	Description
UART_INT	Output	Host Interrupt routed to SERIRQ
EC IF	I/O Bus	Bus used for register access
MCLK	Input	Block operating clock
UART_RX	Input	UART Receive data pin
UART_TX	Output	UART Transmit data pin
UART_CLK	Input	UART Alternate clock pin (1.8462MHz)
nSYS_RST	input	VTR POR reset
nSIO_RESET	input	VCC POR reset

13.2 Power, Clocks and Reset

13.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

13.2.2 CLOCKS

Registers in this block are clocked at the LPC Bus Clock rate which is derived by the MCLK. Baud rates are derived from 1.8432MHz. The 1.8432MHz. is itself derived from either MCLK or sourced from UART_CLK Signal Pin Function.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

In order to maintain communicating with acceptable error, an accurate baud clock is required.

Note 13-1 The FREQ LOCK bit in the PCR Status and Control Register on page 104 must be set in order to insure an accurate baud clock when the CLK_SRC bit is '0' in the Configuration Select Register on page 217, the baud clock is internally sourced.

Note 13-2 When the CLK_SRC bit is '1' in the Configuration Select Register on page 217, the baud clock is externally sourced from the UART_CLK pin. The UART_CLK requires a frequency of 1.8432 MHz ± 2%.

13.2.3 RESET

Table 13-2 details the effect of nSYS_RST or nSIO_RESET on each of the runtime registers of the Serial Port.

TABLE 13-2: RESET FUNCTION TABLE

Register/Signal	Reset Control	Reset State
Interrupt Enable Register	RESET	All bits low
Interrupt Identification Reg.		Bit 0 is high; Bits 1 - 7 low
FIFO Control		All bits low
Line Control Reg.		
MODEM Control Reg.		
Line Status Reg.		All bits low except 5, 6 high
MODEM Status Reg.		Bits 0 - 3 low; Bits 4 - 7 input
TXD1, TXD2		High
INTRPT (RCVR errs)	RESET/Read LSR	Low
INTRPT (RCVR Data Ready)	RESET/Read RBR	
INTRPT (THRE)	RESET/Read IIR/Write THR	
OUT2B	RESET	High
RTSB		
DTRB		
OUT1B		
RCVR FIFO	RESET/ FCR1*FCR0/_FCR0	All Bits Low
XMIT FIFO	RESET/ FCR1*FCR0/_FCR0	

The Runtime register can be configured to be reset on either nSYS_RST or nSIO_RESET. The POWER bit in the Configuration Select Register controls which reset effects the runtime registers. The Refer to Table 13-2 for effected registers and Section 5.0, "Power, Clocks and Resets" for definitions of nSYS_RST on page 98 or nSIO_RESET on page 75.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

13.3 Interrupts

13.3.1 EC INTERRUPT

The Two Pin Serial Port (UART) can generate an EC interrupt event. The interrupt source is routed onto the UART_RX bit in the GIRQ15 Source Register, and is a level sensitive, active high signal.

13.3.2 HOST INTERRUPT

The Two Pin Serial Port (UART) can generate a SIRQ event to the Host. See the Interrupt Enable Register (IER) on page 207 and the Interrupt Identification Register (IIR) on page 208. This interrupt is routed to the SIRQ block. (See SERIRQ Configuration Registers on page 63.)

13.4 Registers

The Two Pin Serial Port (UART) registers are located on the Host SPB.

Each instance of the Two Pin Serial Port (UART) has its own Logical Device Number, and Base Address as indicated in Table 13-3.

TABLE 13-3: Two Pin Serial Port (UART) BASE ADDRESS TABLE

Two Pin Serial Port (UART) Instance	LDN from (Table 3-2 on page 48)	AHB Base Address
UART	7h	FF_1C00h

Note 13-3 The Host LPC I/O addresses for each instance is selected via a Base Address Register (see Section 4.6.2, "Base Address Registers," on page 56). LPC access to configuration registers is through the Host Access Configuration Port. (See Section 4.5.1, "Host Access Port," on page 55.)

Table 13-4 is a register summary for one instance of the Two Pin Serial Port (UART). The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port is via its LDN indicated in Table 13-3 on page 203 and its Host Access Port index which is described as “Host Config Index” in the tables below.

The following table summarizes the registers allocated for the Controller. The offset field in the following table is the off-set from the Embedded Controller’s (EC) Base Address.

TABLE 13-4: Two Pin Serial Port (UART) REGISTER SUMMARY

	Host I/O Access			DLAB (Note 13-4)	EC Interface			
Register Name	**Host I/O Index**	**SPB Off-set**	**Host Type**		**SPB Off-set**	**Byte Lane**	**EC Type**	**Notes**
Receive Buffer Register (RB)	00h	00h	R	0	00h	0	R	Note 13-5
Transmit Buffer Register (TB)	00h	00h	W	0	00h	0	W	Note 13-5
Programmable Baud Rate Generator (and Divisor) **(LSByte)**	00h	00h	R/W	1	00h	0	R/W	Note 13-5
Programmable Baud Rate Generator (and Divisor) **(MSByte)**	01h	01h	R/W	1	01h	1	R/W	Note 13-5
Interrupt Enable Register (IER)	01h	01h	R/W	0	01h	1	R/W	Note 13-5
FIFO Control Register (FCR)	02h	02h	W	X	02h	2	W	Note 13-5
Interrupt Identification Register (IIR)	02h	02h	R	X	02h	2	R	Note 13-5
Line Control Register (LCR)	03h	03h	R/W	X	03h	3	R/W	Note 13-5

TABLE 13-4: Two Pin Serial Port (UART) REGISTER SUMMARY (CONTINUED)

	Host I/O Access			DLAB (Note 13-4)	EC Interface			
Register Name	**Host I/O Index**	**SPB Off-set**	**Host Type**		**SPB Off-set**	**Byte Lane**	**EC Type**	**Notes**
Modem Control Register (MCR)	04h	04h	R/W	X	04h	0	R/W	Note 13-5
Line Status Register (LSR)	05h	05h	R	X	05h	1	R	Note 13-5
Modem Status Register (MSR)	06h	06h	R	X	06h	2	R	Note 13-5
Scratchpad Register (SCR)	07h	07h	R/W	X	07h	3	R/W	Note 13-5
	HOST ACCESS				**EC INTERFACE**			
REGISTER NAME	**HOST CONFIG. INDEX**	**SPB OFF-SET**	**HOST TYPE**	**N/A**	**EC OFFSET**	**BYTE LANE**	**EC TYPE**	
Activate	30h	330h	R/W		330h	0	R/W	Note 13-5
Configuration Select Register	F0h	3F0h	R/W		3F0h	0	R/W	Note 13-5

Note 13-4 DLAB is Bit 7 of the Line Control Register.

Note 13-5 Access to this register should be limited to 8-bit loads and stores. 16-bit or 32-bit stores will be blocked and 16-bit or 32-bit loads have unexpected results. JTAG Debugger access should indirect using peek_poke_arc macros described in IEEE Std 1149.1.

13.5 Register Summary

TABLE 13-5: REGISTER SUMMARY

Address (Note 13-6)	R/W	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADDR = 0 DLAB = 0	R	Receive Buffer r	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0 (Note 13-7)
ADDR = 0 DLAB = 0	W	Transmitter Holding r	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
ADDR = 1 DLAB = 0	R/W	Interrupt Enable r	Reserved				Enable Modem Status Interrupt (EMSI)	Enable Receiver Line Status Interrupt (ELSI)	Enable Transmitter Holding Register Empty Interrupt (ETHREI)	Enable Received Data Available Interrupt (ERDAI)
ADDR = 2	R	Interrupt Ident. r	FIFOs Enabled (Note 13-11)	FIFOs Enabled (Note 13-11)	Reserved		Interrupt ID Bit (Note 13-11)	Interrupt ID Bit	Interrupt ID Bit	"0" if Interrupt Pending
ADDR = 2	W	FIFO Control r	RCVR Trigger MSB	RCVR Trigger LSB	Reserved		DMA Mode Select (Note 13-12)	XMIT FIFO Reset	RCVR FIFO Reset	FIFO Enable
ADDR = 3	R/W	Line Control r	Divisor Latch Access Bit (DLAB)	Set Break	Stick Parity	Even Parity Select (EPS)	Parity Enable (PEN)	Number of Stop Bits (STB)	Word Length Select Bit 1 (WLS1)	Word Length Select Bit 0 (WLS0)
ADDR = 4	R/W	MODEM Control r	**Reserved**			**Loop**	**OUT2 (Note 13-9)**	**OUT1 (Note 13-9)**	**Request to Send (RTS)**	**Data Terminal Ready (DTR)**
ADDR = 5	R/W	Line Status r	**Error in RCVR FIFO (Note 13-11)**	**Transmitter Empty (TEMT) (Note 13-8)**	**Transmitter Holding Register (THRE)**	**Break Interrupt (BI)**	**Framing Error (FE)**	**Parity Error (PE)**	**Overrun Error (OE)**	**Data Ready (DR)**
ADDR = 6	R/W	MODEM Status r	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear to Send (DCTS)
ADDR = 7	R/W	Scratch r (Note 13-10)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDR = 0 DLAB = 1	R/W	Divisor Latch (LS)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDR = 1 DLAB = 1	R/W	Divisor Latch (MS)	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

UART Register Summary Notes:

Note 13-6 DLAB is Bit 7 of the Line Control Register (ADDR = 3).

Note 13-7 Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

Note 13-8 When operating in the XT mode, this bit will be set any time that the transmitter shift register is empty.

Note 13-9 This bit no longer has a pin associated with it.

Note 13-10 When operating in the XT mode, this register is not available.

Note 13-11 These bits are always zero in the non-FIFO mode.

Note 13-12 Writing a one to this bit has no effect. DMA modes are not supported in this chip.

13.6 Detailed Description of Accessible Runtime Registers

13.6.1 RECEIVE BUFFER REGISTER (RB)

TABLE 13-6: RECEIVE BUFFER (RB)

HOST OFFSET	0h (DLAB=0)				8-bit		**HOST SIZE**	
EC OFFSET	0h (DLAB=0)				8-bit		**EC SIZE**	
POWER	VCC or VTR				00h		nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Received Data byte [7:0]							

RECEIVED DATA BYTE

This register holds the received incoming data byte. Bit 0 is the least significant bit, which is transmitted and received first. Received data is double buffered; this uses an additional shift register to receive the serial data stream and convert it to a parallel 8 bit word which is transferred to the Receive Buffer register. The shift register is not accessible.

13.6.2 TRANSMIT BUFFER REGISTER (TB)

TABLE 13-7: TRANSMIT BUFFER (TB)

HOST OFFSET	0h (DLAB=0)				8-bit		**HOST SIZE**	
EC OFFSET	0h (DLAB=0)				8-bit		**EC SIZE**	
POWER	VCC or VTR				00h		nSYS_RST or VCC Power Good **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	W	W	W	W	W	/W	W	W
EC TYPE		W	W	W	W	W	W	W
BIT NAME	Transmit data byte [7:0]							

TRANSMIT DATA BYTE

This register contains the data byte to be transmitted. The transmit buffer is double buffered, utilizing an additional shift register (not accessible) to convert the 8 bit data word to a serial format. This shift register is loaded from the Transmit Buffer when the transmission of the previous byte is complete.

13.6.3 INTERRUPT ENABLE REGISTER (IER)

TABLE 13-8: INTERRUPT ENABLE (IER)

HOST OFFSET	1h (DLAB=0)				8-bit		**HOST SIZE**	
EC OFFSET	1h (DLAB=0)				8-bit		**EC SIZE**	
POWER	VCC or VTR				00h		nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EMSI	ELSI	ETHREI	ERDAI

The lower four bits of this register control the enables of the five interrupt sources of the Serial Port interrupt. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of this register. Similarly, setting the appropriate bits of this register to a high, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and disables any Serial Port interrupt out of the MEC1609/MEC1609i. All other system functions operate in their normal manner, including the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are described below.

ERDAI

This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic "1".

ETHREI

This bit enables the Transmitter Holding Register Empty Interrupt when set to logic "1".

ELSI

This bit enables the Received Line Status Interrupt when set to logic "1". The error sources causing the interrupt are Overrun, Parity, Framing and Break. The Line Status Register must be read to determine the source.

EMSI

This bit enables the MODEM Status Interrupt when set to logic "1". This is caused when one of the Modem Status Register bits changes state.

13.6.4 FIFO CONTROL REGISTER (FCR)

TABLE 13-9: FIFO CONTROL (FCR)

HOST OFFSET	02h				8-bit		**HOST SIZE**	
EC OFFSET	02h				8-bit		**EC SIZE**	
POWER	VCC or VTR				00h		nSYS_RST or VCC Power Good **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	W	W	W	W	W	W	W	W
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	RECV FIFO Trigger Level		Reserved			Clear XMIT FIFO	Clear RECV FIFO	EXRF

This is a write only register at the same location as the IIR.

Note 13-13 DMA is not supported.

EXRF

Enable XMIT and RECV FIFO. Setting this bit to a logic "1" enables both the XMIT and RCVR FIFOs. Clearing this bit to a logic "0" disables both the XMIT and RCVR FIFOs and clears all bytes from both FIFOs. When changing from FIFO Mode to non-FIFO (16450) mode, data is automatically cleared from the FIFOs. This bit must be a 1 when other bits in this register are written to or they will not be properly programmed.

CLEAR RECV FIFO

Setting this bit to a logic "1" clears all bytes in the RCVR FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.

CLEAR XMIT FIFO

Setting this bit to a logic "1" clears all bytes in the XMIT FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.

RECV FIFO TRIGGER LEVEL

These bits are used to set the trigger level for the RCVR FIFO interrupt.

TABLE 13-10: RECV FIFO TRIGGER LEVEL

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

13.6.5 INTERRUPT IDENTIFICATION REGISTER (IIR)

TABLE 13-11: INTERRUPT IDENTIFICATION (IIR)

HOST OFFSET	02h				8-bit		**HOST SIZE**	
EC OFFSET	02h				8-bit		**EC SIZE**	
POWER	VCC or VTR				01h		nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	FIFO_En		Reserved		IntID			IPEND

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

1. Receiver Line Status (highest priority)
2. Received Data Ready
3. Transmitter Holding Register Empty
4. MODEM Status (lowest priority)

Information indicating that a prioritized interrupt is pending and the source of that interrupt is stored in the Interrupt Identification Register (refer to Table 13-12). When the CPU accesses the IIR, the Serial Port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. During this CPU access, even if the Serial Port records new interrupts, the current indication does not change until access is completed. The contents of the IIR are described below.

IPEND

This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic "0", an interrupt is pending and the contents of the IIR may be used as a pointer to the appropriate internal service routine. When bit 0 is a logic "1", no interrupt is pending.

INTID

These three bits of the IIR are used to identify the highest priority interrupt pending as indicated by Table 13-12. In non-FIFO mode, Bit[3] is a logic “0”. In FIFO mode Bit[3] is set along with Bit[2] when a timeout interrupt is pending.

TABLE 13-12: INTERRUPT CONTROL TABLE

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

FIFO_EN

These two bits are set when the FIFO CONTROL Register bit 0 equals 1.

13.6.6 LINE CONTROL REGISTER (LCR)

TABLE 13-13: LINE CONTROL (LCR)

HOST OFFSET	03h			8-bit			**HOST SIZE**	
EC OFFSET	03h			8-bit			**EC SIZE**	
POWER	VCC or VTR			00h			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DLAB	Break Control	Stick Parity	Parity Select	Enable Parity	Stop Bits	Word Length	

This register contains the format information of the serial line. The bit definitions are:

WORD LENGTH

These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

STOP BITS

This bit specifies the number of stop bits in each transmitted or received serial character. Table 13-14 summarizes the information.

TABLE 13-14: STOP BITS

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

Note 13-14 The receiver will ignore all stop bits beyond the first, regardless of the number used in transmitting.

TABLE 13-15: SERIAL CHARACTER

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

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

ENABLE PARITY

Parity Enable bit. When bit 3 is a logic "1", a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and the first stop bit of the serial data. (The parity bit is used to generate an even or odd number of 1s when the data word bits and the parity bit are summed.)

PARITY SELECT

Even Parity Select bit. When bit 3 is a logic "1" and bit 4 is a logic "0", an odd number of logic "1"'s is transmitted or checked in the data word bits and the parity bit. When bit 3 is a logic "1" and bit 4 is a logic "1" an even number of bits is transmitted and checked.

STICK PARITY

Stick Parity bit. When parity is enabled it is used in conjunction with bit 4 to select Mark or Space Parity. When LCR bits 3, 4 and 5 are 1 the Parity bit is transmitted and checked as a 0 (Space Parity). If bits 3 and 5 are 1 and bit 4 is a 0, then the Parity bit is transmitted and checked as 1 (Mark Parity). If bit 5 is 0 Stick Parity is disabled.

Bit 3 is a logic "1" and bit 5 is a logic "1", the parity bit is transmitted and then detected by the receiver in the opposite state indicated by bit 4.

BREAK CONTROL

Set Break Control bit. When bit 6 is a logic "1", the transmit data output (TXD) is forced to the Spacing or logic "0" state and remains there (until reset by a low level bit 6) regardless of other transmitter activity. This feature enables the Serial Port to alert a terminal in a communications system.

DLAB

Divisor Latch Access Bit (DLAB). It must be set high (logic "1") to access the Divisor Latches of the Baud Rate Generator during read or write operations. It must be set low (logic "0") to access the Receiver Buffer Register, the Transmitter Holding Register, or the Interrupt Enable Register.

13.6.7 MODEM CONTROL REGISTER (MCR)

TABLE 13-16: MODEM CONTROL (MCR)

HOST OFFSET	04h			8-bit			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VCC or VTR			00h			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved			LOOP-BACK	OUT2	OUT1	RTS	DTR

This 8-bit register controls the interface with the MODEM or data set (or device emulating a MODEM). The contents of the MODEM control register are described below.

DTR

This bit controls the Data Terminal Ready (nDTR) output. When bit 0 is set to a logic "1", the nDTR output is forced to a logic "0". When bit 0 is a logic "0", the nDTR output is forced to a logic "1".

RTS

This bit controls the Request To Send (nRTS) output. Bit 1 affects the nRTS output in a manner identical to that described above for bit 0.

OUT1

This bit controls the Output 1 (OUT1) bit. This bit does not have an output pin and can only be read or written by the CPU.

OUT2

Output 2 (OUT2). This bit is used to enable an UART interrupt. When OUT2 is a logic "0", the serial port interrupt output is forced to a high impedance state - disabled. When OUT2 is a logic "1", the serial port interrupt outputs are enabled.

LOOPBACK

This bit provides the loopback feature for diagnostic testing of the Serial Port. When bit 4 is set to logic "1", the following occur:

1. The TXD is set to the Marking State (logic "1").
2. The receiver Serial Input (RXD) is disconnected.
3. The output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input.
4. All MODEM Control inputs (nCTS, nDSR, nRI and nDCD) are disconnected.
5. The four MODEM Control outputs (nDTR, nRTS, OUT1 and OUT2) are internally connected to the four MODEM Control inputs (nDSR, nCTS, RI, DCD).
6. The Modem Control output pins are forced inactive high.
7. Data that is transmitted is immediately received.

This feature allows the processor to verify the transmit and receive data paths of the Serial Port. In the diagnostic mode, the receiver and the transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

13.6.8 LINE STATUS REGISTER (LSR)

TABLE 13-17: LINE STATUS (LSR)

HOST OFFSET	05h					8-bit	**HOST SIZE**	
EC OFFSET	05h					8-bit	**EC SIZE**	
POWER	VCC or VTR					60h	nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	FIFO Error	Trans-mit Error	Trans-mit Empty	Break Interrupt	Frame Error	Parity Error	Overrun Error	Data Ready

DATA READY

Data Ready (DR). It is set to a logic "1" whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to a logic "0" by reading all of the data in the Receive Buffer Register or the FIFO.

OVERRUN ERROR

Overrun Error (OE). Bit 1 indicates that data in the Receiver Buffer Register was not read before the next character was transferred into the register, thereby destroying the previous character. In FIFO mode, an overrun error will occur only when the FIFO is full and the next character has been completely received in the shift register, the character in the shift register is overwritten but not transferred to the FIFO. The OE indicator is set to a logic "1" immediately upon detection of an overrun condition, and reset whenever the Line Status Register is read.

PARITY ERROR

Parity Error (PE). Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. The PE is set to a logic "1" upon detection of a parity error and is reset to a logic "0" whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO.

FRAME ERROR

Framing Error (FE). Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to a logic "1" whenever the stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level). The FE is reset to a logic "0" whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. The Serial Port will try to resynchronize after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this 'start' bit twice and then takes in the 'data'.

BREAK INTERRUPT

Break Interrupt (BI). Bit 4 is set to a logic “1” whenever the received data input is held in the Spacing state (logic “0”) for longer than a full word transmission time (that is, the total time of the start bit + data bits + parity bits + stop bits). The BI is reset after the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. Restarting after a break is received, requires the serial data (RXD) to be logic “1” for at least 1/2 bit time.

Bits 1 through 4 are the error conditions that produce a Receiver Line Status Interrupt BIT 3

Note 13-15 whenever any of the corresponding conditions are detected and the interrupt is enabled

TRANSMIT EMPTY

Transmitter Holding Register Empty (THRE). Bit 5 indicates that the Serial Port is ready to accept a new character for transmission. In addition, this bit causes the Serial Port to issue an interrupt when the Transmitter Holding Register interrupt enable is set high. The THRE bit is set to a logic "1" when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic "0" whenever the CPU loads the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty, it is cleared when at least 1 byte is written to the XMIT FIFO. Bit 5 is a read only bit.

TRANSMIT ERROR

Transmitter Empty (TEMT). Bit 6 is set to a logic "1" whenever the Transmitter Holding Register (THR) and Transmitter Shift Register (TSR) are both empty. It is reset to logic "0" whenever either the THR or TSR contains a data character. Bit 6 is a read only bit. In the FIFO mode this bit is set whenever the THR and TSR are both empty.

FIFO ERROR

This bit is permanently set to logic "0" in the 450 mode. In the FIFO mode, this bit is set to a logic "1" when there is at least one parity error, framing error or break indication in the FIFO. This bit is cleared when the LSR is read if there are no subsequent errors in the FIFO.

13.6.9 MODEM STATUS REGISTER (MSR)

TABLE 13-18: MODEM STATUS (MSR)

HOST ADDRESS	06h			8-bit			**HOST SIZE**	
EC OFFSET	06h			8-bit			**EC SIZE**	
POWER	VCC or VTR			xxxx0000b			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	DCD#	RI#	DSR	CTS	DCD	RI	DSR	CTS

This 8 bit register provides the current state of the control lines from the MODEM (or peripheral device). In addition to this current state information, four bits of the MODEM Status Register (MSR) provide change information.

These bits are set to logic "1" whenever a control input from the MODEM changes state. They are reset to logic "0" whenever the MODEM Status Register is read.

CTS

Delta Clear To Send (DCTS). Bit 0 indicates that the nCTS input to the chip has changed state since the last time the MSR was read.

DSR

Delta Data Set Ready (DDSR). Bit 1 indicates that the nDSR input has changed state since the last time the MSR was read.

RI

Trailing Edge of Ring Indicator (TERI). Bit 2 indicates that the nRI input has changed from logic "0" to logic "1".

DCD

Delta Data Carrier Detect (DDCD). Bit 3 indicates that the nDCD input to the chip has changed state.

Note 13-16 Whenever bit 0, 1, 2, or 3 is set to a logic "1", a MODEM Status Interrupt is generated.

CTS

This bit is the complement of the Clear To Send (nCTS) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to nRTS in the MCR.

DSR

This bit is the complement of the Data Set Ready (nDSR) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to DTR in the MCR.

RI#

This bit is the complement of the Ring Indicator (nRI) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to OUT1 in the MCR.

DCD

This bit is the complement of the Data Carrier Detect (nDCD) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to OUT2 in the MCR.

APPLICATION NOTE: The Modem Status Register (MSR) only provides the current state of the UART MODEM control lines in Loopback Mode. The MEC1609/MEC1609i does not support external

connections for the MODEM Control inputs (nCTS, nDSR, nRI and nDCD) or for the four MODEM Control outputs (nDTR, nRTS, OUT1 and OUT2).

13.6.10 SCRATCHPAD REGISTER (SCR)

TABLE 13-19: SCRATCH PAD (SCR)

HOST OFFSET	07h					8-bit	**HOST SIZE**	
EC OFFSET	07h					8-bit	**EC SIZE**	
POWER	VCC or VTR					00h	nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Scratch							

SCRATCH

This 8 bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

13.6.11 PROGRAMMABLE BAUD RATE GENERATOR (AND DIVISOR)

TABLE 13-20: PROGRAMMABLE BAUD RATE GENERATOR (AND DIVISOR)

HOST OFFSET	BYTE1: 01h (DLAB = 1) BYTE0: 00h (DLAB = 1)					8-bit	**HOST SIZE**	
EC OFFSET	BYTE1: 01h (DLAB = 1) BYTE0: 00h (DLAB = 1)					8-bit	**EC SIZE**	
POWER	VCC or VTR					0000h	nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Baud_Clock_Sel	Baud_Rate_Divisor						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Baud_Rate_Divisor[7:0]							

BAUD_CLOCK_SEL

If the CLK_SRC bit is '0' and the Baud_Clock_Sel bit is '0,' the 1.8432MHz clock is used to generate the baud clock. Table 13-21 shows some baud rates that can be generated with this clock. The CLK_SRC bit is D0 in the UART Logical Device configuration register offset 0xF0.

If the CLK_SRC bit is ‘0’ and the Baud_Clock_Sel bit is ‘1,’ the 32.26MHz clock is used to generate the baud clock. Table 13-22 shows some baud rates that can be generated with this clock.

If the CLK_SRC bit is ‘1,’ the Baud_Clock_Sel bit as no effect.

BAUD_RATE_DIVISOR

The Serial Port contains a programmable Baud Rate Generator that is capable of dividing the internal clock source by any divisor from 1 to 65535. The clock source is either a 1.8432MHz clock derived from the 64.52MHz ring oscillator or a 32.26Mhz clock also derived from the ring oscillator. The output frequency of the Baud Rate Generator is 16x the Baud rate. Two eight bit latches store the divisor in 16 bit binary format. These Divisor Latches must be loaded during initialization in order to insure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16 bit Baud counter is immediately loaded. This prevents long counts on initial load. If a 0 is loaded into the BRG registers, the output divides the clock by the number 3. If a 1 is loaded, the output is the inverse of the input oscillator. If a two is loaded, the output is a divide by 2 signal with a 50% duty cycle. If a 3 or greater is loaded, the output is low for 2 bits and high for the remainder of the count.

Table 13-21 and Table 13-22 shows the baud rates possible.

TABLE 13-21: UART BAUD RATES (1.8432MHZ SOURCE)

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

TABLE 13-22: UART BAUD RATES (32.26MHZ SOURCE)

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
126000	1	16
168000	1	12
183000	1	11
201600	1	10
224000	1	9
252000	1	8
288000	1	7
336000	1	6

TABLE 13-22: UART BAUD RATES (32.26MHZ SOURCE) (CONTINUED)

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
403800	1	5
504100	1	4
672100	1	3
1008000	1	2
2016000	1	1

13.7 Detailed Description of Configuration Registers

13.7.1 ACTIVATE

TABLE 13-23: ACTIVATE REGISTER

HOST OFFSET	30h					8-bit	**HOST SIZE**	
EC OFFSET	330h					32-bit	**EC SIZE**	
POWER	VTR					00b	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the UART logical device is powered and functional. When this bit is 0, the UART logical device is powered down and inactive.

13.7.2 CONFIGURATION

TABLE 13-24: CONFIGURATION SELECT REGISTER

HOST OFFSET	F0h					8-bit	**HOST SIZE**	
EC OFFSET	3F0h					8-bit	**EC SIZE**	
POWER	VTR					00b	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved					Polarity	Power	CLK_SRC

CLK_SRC

When this bit is 0, the UART clock is derived from the internal ring oscillator. When this bit is 1, the UART clock is derived from an external clock source.

POWER

When this bit is 1, the UART Runtime Registers (the registers at offsets 0h through 7h from the base of the UART Logical Device) are controlled by VCC. They are set to their POR defaults on a nSIO_RESET. In addition, pins associated with the UART are powered down and place in a High-Z state on nSIO_RESET.

When this bit is 0, the UART Runtime Registers are controlled by VTR. They are set to their POR defaults on an nSYS-_RST. In addition, the state of the UART pins is controlled by VTR.

POLARITY

When the Polarity bit is asserted ('1'), the UART_TX and UART_RX pins functions are inverted. When the Polarity bit is not asserted (default), the UART_TX and UART_RX pins functions are not inverted.

13.8 Sleep Enable/ Clock Request Power state controls

TABLE 13-25: UART BLOCK CLOCK GATING BEHAVIOR

Activate	External Sleep Input	Block Idle Status (Note 13-17)	Clock Required Status Output	State	Description
0	X	X	0	DISABLED	Two Pin Serial Port (UART) is disabled by firmware and the core clock is not needed. Note: it is up to the host to ensure that the block is not in use before the Activate bit is de-asserted.
1	0	NOT IDLE	1	NORMAL OPERATION	The block is neither disabled by firmware nor commanded to sleep.
		IDLE	0		
	1	NOT IDLE	1	PREPARING TO SLEEP	A sleep command has been asserted but the core clock is still required because the block is not idle.
		IDLE	0	SLEEPING	A sleep command has been asserted, the block is idle and the core clock can be stopped.

Note 13-17 the Two Pin Serial Port (UART) 'idle' status is defined in Table 13-26.

TABLE 13-26: UART IDLE STATUS

Transmitter Active?	Receiver Active?	Character Time-out Active	Status
NO	NO	NO	Idle
YES	X	X	Not Idle
X	YES	X	
X	X	YES	

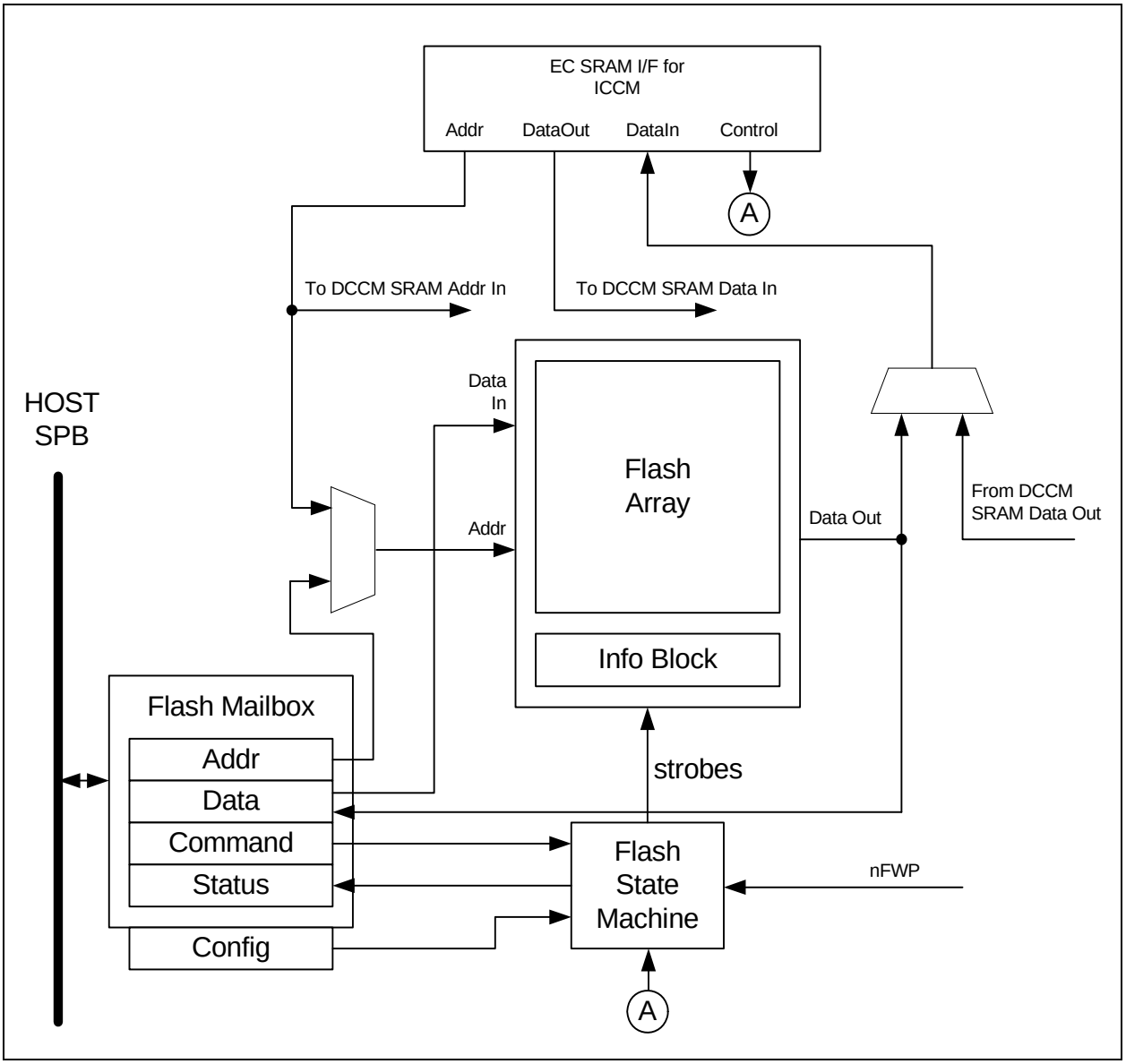
14.0 EMBEDDED FLASH SUBSYSTEM

14.1 General Description

The MEC1609/MEC1609i Embedded Flash Subsystem includes a 192KB embedded Flash memory. The memory appears in the system AHB address space and can store both instructions and data. The Flash memory can be programmed by the Embedded Controller, by the Host via LPC, and by ATE via JTAG.

14.2 Block Diagram

FIGURE 14-1: EMBEDDED FLASH BLOCK DIAGRAM



14.3 Power, Clocks and Reset

14.3.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

14.3.2 CLOCKS

This block uses the LPC Bus Clock, the 64.52MHz MCLK and the 100KHz MCLK_DIV640_EN. LPC Bus Clock is used when reading and writing the Embedded Flash Subsystem control registers. All Flash signal timing is derived from the MCLK. The 100KHz MCLK_DIV640_EN is used to determine the multiple-millisecond times required for Erase Mode.

The Embedded Flash controller will keep the internal ring oscillator operating as long as the controller is not in the Standby state. This permits the controller to complete any program or erase operation even though the Embedded Controller may be in its sleep state.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

14.3.2.1 Clock Idle

The Embedded Flash controller will keep the internal ring oscillator operating as long as the controller is not in the Standby state. This permits the controller to complete any program or erase operation even though the Embedded Controller may be in its sleep state. See APPLICATION NOTE: on page 231.

14.3.3 RESET

This block is reset by nSYS_RST. Following a reset, all registers are set to their default values, and the internal state machines are reset to the standby state.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

The JTAG interface registers referenced in this chapter have an asynchronous reset. See Note 39-1 on page 484.

14.3.4 TRACKING FLASH PROGRAM OR ERASE ACTIVITY

When the Embedded Flash Command Register is placed in Program Mode or Erase Mode, the FLASH bit is asserted in the Power-Fail and Reset Status Register. The FLASH bit in the Power-Fail and Reset Status Register is sticky, VBAT powered and reset by VBAT_POR.

APPLICATION NOTE: The purpose of the FLASH bit in the Power-Fail and Reset Status Register is to provide the EC with status. Once this bit is asserted by hardware, EC firmware should clear the FLASH bit as soon as possible after every Program Mode or Erase Mode operation. EC software can detect unexpected events which may indicate flash corruption. For example, VTR POR, VCC POR, LRESET#, and VCC_PWRGD transitions, as well as, completion of host flash erase or programming are example events when firmware should examining & clear the FLASH bit in the Power-Fail and Reset Status Register can be useful.

If a reset occurs, the software reset handler can examine the FLASH bit to determine if the Flash memory might be corrupted. Corruption can occur if Flash programming or erasure is interrupted by a reset.

If VCC power is removed, or if LRESET# is asserted, the LPC interface becomes inactive and the Embedded Flash controller must ensure that the flash subsystem is in a consistent state. When the LPC bus is inoperative the EC should have full access to the Embedded Flash. Table 14-1, "VCC PWRGD and LRESET# Behavior" describes the different possible states and the response.

TABLE 14-1: VCC PWRGD AND LRESET# BEHAVIOR

Embedded Flash Configuration Register	Embedded Flash Command Register	Action on LRESET# Asserted or VCC PWRGD De-Asserted
Host_Ctl	Flash_Mode	
0	X	No action (see Note 14-1)
1	Standby Mode	• Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. • EC_Int in Embedded Flash Command Register is set to '1b'. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.
1	Read Mode	• Any read of the Embedded Flash array in progress is completed. • All strobes to the Embedded Flash array are set to '0b' • Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. The Program Mode is set to Standby. • EC_Int in Embedded Flash Command Register is set to '1b'. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.
1	Program Mode	• Any word currently being programmed to the Embedded Flash array in progress is completed. • The Program Mode epilogue sequence is issued. • All strobes to the Embedded Flash array are set to '0b' • Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. The Program Mode is set to Standby. • EC_Int in Embedded Flash Command Register is set to '1b'. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.
1	Erase Mode	• The erase sequence is completed. • All strobes to the Embedded Flash array are set to '0b' • Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. • EC_Int in Embedded Flash Command Register is set to '1b'. The Program Mode is set to Standby. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.

Note 14-1 The assertion of LRESET# or the de-assertion of VCC PWRGD has no effect in this state; the EC can set or clear Reg_Ctl_En, Reg_Ctl and Host_Ctl and EC_Int is not set.

APPLICATION NOTE: The FLASH bit in the Power-Fail and Reset Status Register on page 111 is asserted autonomously and is sticky. Firmware should examine this bit after either VCC POR or LRESET# assertion.

14.4 Interrupts

The Embedded Flash Subsystem can generate interrupts to the EC for four events, three of which are reported in the three error bits in the Embedded Flash Status Register: Protect_Err, CMD_Err, & Busy_Err. The error bits are routed onto the FLASH_PROTECT_ERR, FLASH_CMD_ERR, FLASH_BUSY_ERR bits of the GIRQ14 Source Register on page 274.

In addition, asserting EC_Int in the Embedded Flash Command Register can be used to generate an interrupt to the EC. This bit are routed onto the FLASH_EC_INT, bit of the GIRQ14 Source Register on page 274.

EC interrupts generated by the LRESET# and VCC PWRGD pin signals are utilized as part of algorithms described in Section 14.10, "Programming the Embedded Flash Array," on page 231. The LRESET# pin signal sets the LRESET# bit in the GIRQ14 Source Register on page 274. The VCC PWRGD pin signal sets the GPIO057 bit interrupt in the GIRQ10 Source Register on page 267.

14.5 Flash Memory Array

The 192kB Flash Memory Array is composed of 48K x 32-bit use array and one 512 x 32-bit information arrays called Non-Volatile Register (NVR) block. In discussions pertaining to flash, the terms page and block are used interchangeably - both refer to contiguous array of 512 32-bit words.

An erase operation in the Flash Memory Array sets the affected memory array bits to one, while program operations write zeros. To reprogram any ‘0’ bit in a page to ‘1,’ the page must be erased.

The Flash Memory Array erases and programs with a 3.3V power supply; its IO interface operates at 1.8V. To modify the contents of the Flash Memory Array, VTR must be >3V before program or erase operations may begin. A summary of the MEC1609/MEC1609i Flash Memory Array features is shown below in Table 14-2.

TABLE 14-2: MEC1609/MEC1609I 192K FLASH FEATURE SUMMARY

Feature		Description
PROG/ERASE VOLTAGE		3.3V ± 10% (T_J = 0×C to 125×C)
READ VOLTAGE		1.8V ± 10% (T_J = 0×C to 125×C)
BUS WIDTH		32-bit
READ ACCESS/CYCLE TIME		35 ns
MEMORY ARRANGEMENT	MAIN BLOCK	48k x 32 in 96 pages of 2048 bytes
	INFO. BLOCK	512 x 32 in one page of 2048 bytes
BOOT BLOCK	SIZE	4096 bytes
	LOCATION	Bottom
ERASE	TYPES	Page/Mass (2048 bytes/page)
	PROGRAM/ ERASE CYCLES	1,000 Cycles
DATA RETENTION TIME		Greater than 10 years at room temperature
PROGRAMMING		Per 32-bit word
INTERFACE		All Program and Erase Operations are Enabled via a Command Sequence Interface using the Embedded Flash Register Interface.

14.5.1 FLASH ADDRESS MAPPING

The Main Block of the Flash Memory Array is located at 00_0000h in the EC address space. Address greater than 192K will wrap around. The 2KB Info block is located at 4_0000h. All locations can be used for both program and data by the EC. See FIGURE 3-2: MEC1609/MEC1609i EC Memory Map on page 47.

The control registers used for programming the Flash are part of the Embedded Flash Logical Device. They are located in AHB address space starting at location FF_3800h. The contents of both the Main Block and the Info block can also be read using the control registers.

14.6 Instruction Memory Interface

When Reg_Ctl of the Embedded Flash Command Register is ‘0b’, access to the Embedded Flash memory is through the Instruction Closely Coupled Memory interface of the EC. The registers in the Register Interface may be read or written, but no operations will be initiated on the Embedded Array. The Flash memory can be read with no wait states at the peak EC clock rate of 21.5MHz; therefore, the EC Clock Divider Register on page 102 must have a value equal to or greater than 3h.

The minimum flash read access time through the EC’s Instruction Closely Coupled Memory interface is 46.5ns = 3 MCLKs.

14.7 Register Interface

Each instance of the Embedded Flash Subsystem has it own Logical Device Number, and Base Address as indicated in Table 14-3.

TABLE 14-3: Embedded Flash Subsystem BASE ADDRESS TABLE

Embedded Flash **Instance**	**LDN from (**Table 3-2 on page 48**)**	**AHB Base Address**
Embedded Flash	Eh	FF_3800h

Table 14-4, “Embedded Flash Subsystem Register Summary,” on page 224 summarizes the registers allocated for the Embedded Flash Subsystem.

The Table 14-4 is a register summary for one instance of the Embedded Flash Subsystem. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address.

Both the Host and the Embedded Controller can communicate with the Embedded Flash Subsystem through a set of registers located on the LPC SPB bus. The Host uses the two byte Flash Mailbox interface, while the EC can access the register interface described in Section 14.7.2, "EC Register Interface".

14.7.1 FLASH MAILBOX INTERFACE

The Flash Mailbox interface provides the Host with a Message Interface similar to the MailBox Register Interface (see Section 12.0). The Flash Mailbox interface port occupies two addresses in the Host I/O space: FL_MBX_INDEX Register & FL_MBX_DATA Register.

To access a Flash Mailbox register once the Flash Mailbox BAR has been initialized, write the Host Offset of the desired 8-bit register into the Flash Mailbox INDX register. The register can then be accessed by reading or writing the Flash Mailbox DATA register.

It consists of 16 index-addressable 8-bit registers. These registers correspond to the four 32-bit registers that are shared between the Host and the EC.

To access a Flash Mailbox register once the Flash Mailbox BAR has been initialized, write the Host Offset of the desired 8-bit register into the Flash Mailbox INDX register. The register can then be accessed by reading or writing the Flash Mailbox DATA register.

Note 14-2 In this specification, Host access to registers in the Flash Mailbox through the Flash Mailbox Access Port are identified by the prefix INDX in OFFSET fields in the register tables in Section 14.11, "Detailed Description of Accessible Registers".

14.7.2 EC REGISTER INTERFACE

The following table summarizes the registers allocated for the Embedded Flash Subsystem. The offset field in the following table is the offset from the EC Base Address.

TABLE 14-4: EMBEDDED FLASH SUBSYSTEM REGISTER SUMMARY

	Host I/O Access			EC Interface		
Flash Mailbox interface **Register Name**	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**EC Type**	**VT (nSYS_RST)**
FL_MBX_INDEX Register	00h	00h	R/W	00h	R/W	00h
FL_MBX_DATA Register	04h	04h	R/W	04h	R/W	00h
Register Name	**Flash Mailbox Index**		**Host Type**	**SPB Offset**	**EC Type**	**VTR POR**
Embedded Flash Data Register	INDX 00h, 01h, 02,h, 03h	-	R/W	100h	R/W	0000_0000h
Embedded Flash Address Register	INDX 04h, 05h, 06h,	-	R/W	104h	R/W	0000_0000h
Embedded Flash Command Register	INDX 08h, 09h	-	R/W	108h	R/W	0000_0000h
Embedded Flash Status Register	INDX 0Ch, 0Dh	-	R	10Ch	R	0000_0000h
Embedded Flash Configuration Register	N/A	-	-	110h	R/W	0000_0000h
Embedded Flash Initialization Register	N/A	-	-	114h	R/W	0000_0000h

14.7.3 FLASH ADDRESS AND DATA REGISTERS

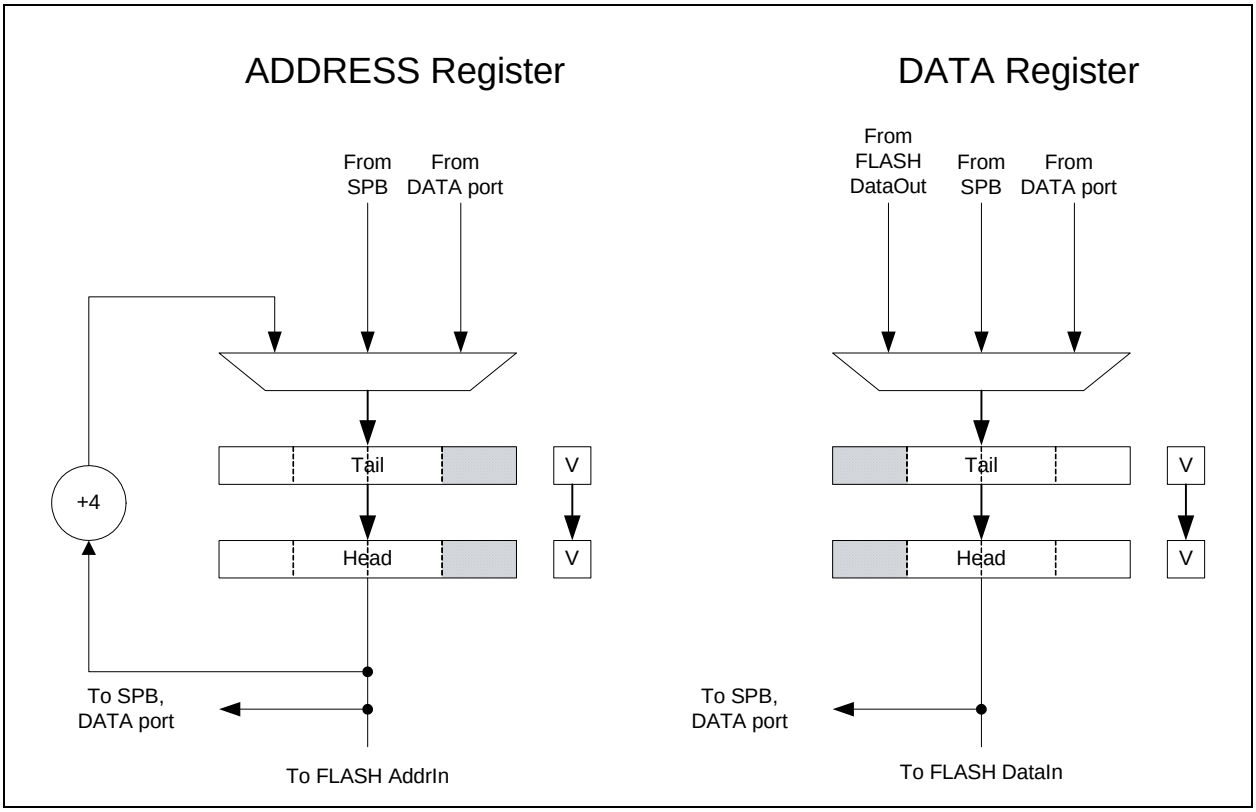
The Embedded Flash Data Register and the Embedded Flash Address Register are each implemented as a two entry FIFO, as illustrated in Figure 14-2, "Embedded Flash Controller Address and Data Registers". The figure is suggestive and does not represent the precise implementation. Writes are always directed to the Tail register of the FIFO and reads are always sourced from the Head register. Each of the two registers has an associated Valid bit (**V** in the figure). The Tail Valid bit is set when the full Tail register is written and cleared when the Tail register is copied into the Head register. The Head Valid bit of the Embedded Flash Data Register is cleared whenever the full Head register is read by the Host or EC, or when the Embedded Flash Controller completes a Program transaction.The Head Valid bit of the Embedded Flash Address Register is cleared whenever the Embedded Flash controller completes a Read, Program or Erase function. Reading the Embedded Flash Address Register by a Host or EC register read does not affect the Head Valid bit. If the Head Valid bit is cleared and the Tail Valid bit is set, the Tail register is copied into the Head register, the Tail Valid bit is cleared and the Head Valid bit is set. All Valid bits are cleared when Flash_Mode of the Embedded Flash Command Register is set to Standby Mode.

The Embedded Flash controller uses byte 0 of the Embedded Flash Address Register and byte 3 of the Embedded Flash Data Register to determine when the full register has been read or written. Writing byte 0 of the Embedded Flash Address Register sets the Valid bit on the Tail of the Address FIFO, while writing byte 3 of the Embedded Flash Data Register sets the Valid bit for the Tail of the Data FIFO. Reading byte 3 of the Embedded Flash Data Register clears the Valid bit for the Head of the Data FIFO. As stated above, reading any byte of the Embedded Flash Address Register does not cause a change in the Head Valid bit.

When the Valid bits of both the Head and Tail of the Embedded Flash Data Register are set, Data_Full in the Embedded Flash Status Register is set. If either bit is cleared, Data_Full is cleared. When the Valid bits of both the Head and Tail of the Embedded Flash Address Register are set, Address_Full in the Embedded Flash Status Register is set. If either bit is cleared, Address_Full is cleared.

The requesting Master will be stalled while attempting to read the Embedded Flash Data Register if the Embedded Flash controller is Busy and in the process of reading data into the Embedded Flash Data Register. This will result in at most one EC wait state. Because the Host cannot read the Flash registers faster than the controller can read from the Flash array, a Host Read access over the LPC bus will not have any added Wait SYNC cycles because the Flash controller is busy. In other cases, a Host or EC read of either the Embedded Flash Data Register or the Embedded Flash Address Register always returns the value in the Head register, whether the Valid bit is set or not.

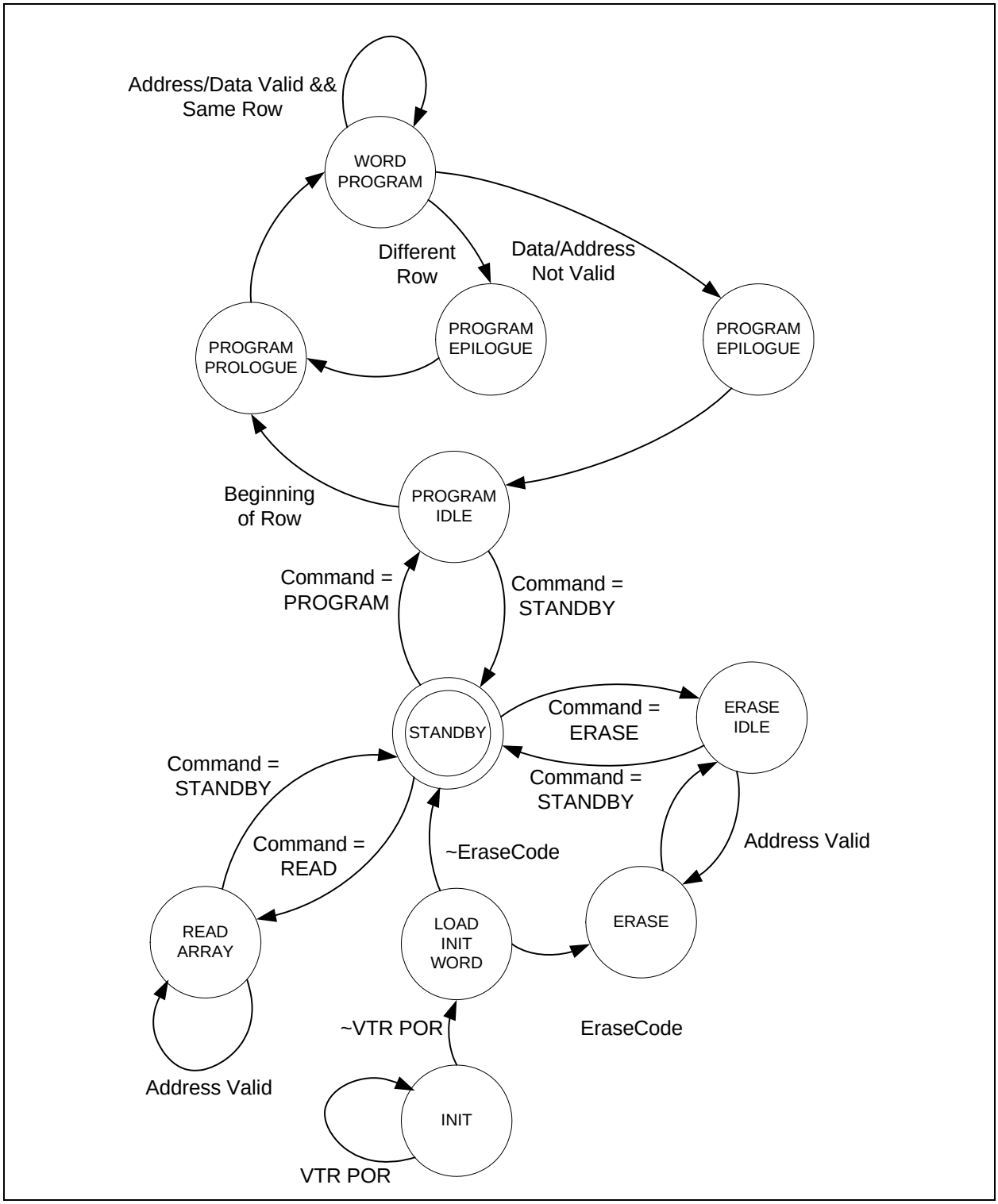
FIGURE 14-2: EMBEDDED FLASH CONTROLLER ADDRESS AND DATA REGISTERS



14.8 Embedded Flash Controller State Sequencing

The Embedded Flash controller proper timing on all strobe signals of the Embedded Flash array. Figure 14-3, "Embedded Flash Controller State Diagram" illustrates the primary state transitions for the controller.

FIGURE 14-3: EMBEDDED FLASH CONTROLLER STATE DIAGRAM



During VTR Reset, the Embedded Flash controller is kept in an initial state (INIT). In normal operation, when VTR POR (nSYS_RST) is de-asserted, the Embedded Flash controller reads the initialization value into the Embedded Flash Initialization Register, as described in Section 14.9.4, "Flash Data Initialization," on page 230, after which it remains in Standby Mode. If the ME (Mass Erase) field in the JTAG TEST REGISTER 4/Reset Register (Dh) is '1b' when VTR POR (nSYS_RST) is de-asserted the Embedded Flash controller, after transitioning to the Flash Data Initialization state will transition to Erase Mode. See Section 14.9.5, "Emergency Mass Erase," on page 230 for detailed sequence and description.

After initialization during normal operation, the Embedded Flash controller is placed into one of four modes (Standby Mode, Read Mode, Program Mode or Erase Mode) by setting Flash_Mode of the Embedded Flash Command Register to the appropriate values (0, 1, 2 or 3, respectively). Any write to the Embedded Flash Command Register while the controller is busy (when Busy of the Embedded Flash Status Register is '1b') will not modify the state and will set Busy_Err in the Embedded Flash Status Register.

14.8.1 STANDBY MODE

When in this mode all strobes to the Flash Memory Array are de-asserted and the Flash Memory Array is placed in its lowest power state. Both the Embedded Flash Data Register FIFO and the Embedded Flash Address Register FIFO are flushed. The registers can be read and written by software without error. Data returned on a read of a flushed FIFO is undefined. On a transition from Standby Mode to Program Mode, Read Mode or Erase Mode the FIFOs will be invalid and no action dependent on valid data in a FIFO will take place until new data is written into the registers.

The Embedded Flash controller must return to Standby Mode after any other state. An attempt to set the controller into a state other than Standby Mode when the controller is in Program Mode, Read Mode or Erase Mode will put the controller in Standby Mode and set CMD_Err of the Embedded Flash Status Register. Flash_Mode of the Embedded Flash Command Register will also be left in Standby Mode. If the controller is busy, writes to the Embedded Flash Command Register are ignored and Busy_Err in the Embedded Flash Status Register is set.

14.8.2 READ MODE

When the Embedded Flash controller is in Read Mode the addressing strobes are kept de-asserted as long as no Read is in progress. When the strobes are de-asserted, the Embedded Flash array is maintained in its lowest power state (equivalent to Standby Mode). When the Valid bit associated with the Head register in the Embedded Flash Address Register is '1b' and the Data FIFO is not full, the data in the Embedded Flash array that corresponds to the Embedded Flash Address Register is loaded into the Tail register of the Embedded Flash Data Register. The Data FIFO can hold two read results.

The Address FIFO is advanced when the Embedded Flash controller completes a read from the Embedded Flash array and stores the result in the Data FIFO. When byte 3 (the most significant byte) of the Embedded Flash Data Register is read, the Data FIFO is advanced. A read of the will return the value of the Head register but will not advance the FIFO. A read of the Embedded Flash Data Register when the FIFO is empty but the controller is busy reading from the array will stall the read until the controller has completed the lookup and the data can be returned to the requestor.

14.8.2.1 Read Mode Timing Parameters

The minimum flash read access time through the EC Register Interface is 46.5ns = 3*MCLK.

14.8.2.2 Burst Read Mode

When Burst in the Embedded Flash Command Register is set to '1b', the Read function automatically increments the Embedded Flash Address Register in order to minimize the time necessary to read a block of memory from the Embedded Flash. As in Read Mode, a Flash read is initiated when the Head of the Address FIFO is Valid and the Data FIFO is empty. Whenever the Tail register in the Address FIFO is not Valid and the Head register in the FIFO is Valid, the Head register is incremented by 4 and written into the Tail. The Address FIFO will thus contain a sequence of consecutive word addresses as long as the Embedded Flash controller remains in Burst Read Mode.

Reading the Embedded Flash Data Register from either the EC or the JTAG interface always reads 32 bits at a time, so the Data Register FIFO is always advanced when the EC or JTAG does a data read. In Burst Read Mode the entire Flash memory can be read by repeatedly reading the Embedded Flash Data Register without the need for reading or writing any other register in the Embedded Flash Subsystem.

The Flash Mailbox interface has an additional mechanism. When in Burst Read Mode, the INDX portal has special behavior when it has the value 0h through 3h (that is, when INDX is set to point to the Embedded Flash Data Register). Every time the DATA portal is read by the Host, INDX will be automatically incremented by 1 when its value is 0h through 2h. If DATA is read by the Host when INDX is 3h, INDX is set to 0h and the Valid bit of the Head register of the Data

FIFO is cleared. If there is a valid address in the Address FIFO, the sequence to read the next data value, described above, will be initiated. When in Burst Read Mode, the entire contents of the Flash memory can be read by the Host with a sequence of reads to DATA, without any intervening writes to INDX.

APPLICATION NOTE: In Burst Read Mode the Embedded Flash Data Register will always contain the data from the next two locations after the last data location that is read. Software should ensure that the last word read does not cause an unintended read into a protected region. For example, a Burst Read that is intended to read the last word before the Protected Data Region will attempt to read into the protected region, which may cause an unintended protection error. To read the last data before the protected region, software should turn off Burst Read Mode and return to Standby Mode two words before the end of the page before the protected region. It should then re-enter Read Mode and read the last two words by explicitly writing the addresses into the Embedded Flash Address Register.

14.8.3 PROGRAM MODE

When the Embedded Flash controller is in Standby Mode, setting the Embedded Flash Command Register to Program Mode will set up the Embedded Flash array for programming. The Embedded Flash Data Register and the Embedded Flash Address Register FIFOs are used for Program Mode in a manner similar to their use in Read Mode. When the Head registers in both the Address and Data FIFO are Valid, the Flash strobes are sequenced in order to write the contents of the Data register into the Flash Memory at the address specified in the Address register. At the end of the Word Program sequence the two FIFOs are advanced.

14.8.3.1 Program Mode Timing Parameters

The following three sequences are used in programming the Embedded Flash:

1. Program Prologue: the controller issues a three-byte load sequence for Software Data Protection, i.e., specific byte patterns onto the flash address and data buses.
2. Word Program: the controller issues the strobes required to program the 32 bits present in the Head register of the Data FIFO into the Embedded Flash array at the address specified in the Head register of the Address FIFO. If Address bit 18 is ‘1b’, the Data word is programmed into the Info block; if Address bit 18 is ‘0b’, the Data word is programmed into the main Flash array.
3. Program Epilogue: the controller issues the strobes to initiate Program operation inside the memory core.

During all three sequences, the controller asserts Busy while the sequence is in progress.

It takes approximately 20.7 µs to program one word.

The Program Prologue is issued before the first time any word in a row can be programmed. The Program Epilogue is issued after the last time any word of a row is programmed. The Embedded Flash controller automatically issues the Program Prologue and Program Epilogue sequences as required. After the controller is placed in Program Mode, the Program Prologue is issued as soon as the Head registers in both the Embedded Flash Data Register and the Embedded Flash Address Register are Valid. Once the Program Prologue is completed, the controller will immediately issue the Word Program sequence.

As long as the Head registers in both the Embedded Flash Data Register and the Embedded Flash Address Register are Valid the controller will continue to issue Word Program sequences. If either FIFO is invalid at the end of the Word Program sequence, the controller will issue the Program Epilogue sequence and the controller will be left in the initial Program Mode state. If new data arrives in the FIFOs, then the Program Prologue followed by the Word Program sequence will be issued.

14.8.3.2 Burst Program Mode

Burst Program Mode is enabled whenever Burst in the Embedded Flash Command Register is set and the controller is in Program Mode. The behavior is similar to Burst Read Mode. When Burst is enabled, the Embedded Flash Address Register FIFO is always kept filled automatically with incrementing addresses. Whenever the Tail register in the Address FIFO is not Valid and the Head register is valid, the Head register is incremented by 4 and stored in the Tail. The controller otherwise behaves as in Program Mode. When both the Head register of the Address FIFO and the Head register of the Data FIFO are Valid, a Word Program sequence is initiated (with the possible addition of Program Prologue and Program Epilogue, as described above). When the programming sequence is complete, the Head registers of both FIFOs are marked not Valid. With this mechanism the entire Flash array can be programmed with a sequence of writes to the Embedded Flash Data Register without any writes to the Embedded Flash Address Register or Embedded Flash Command Register.

The Flash Mailbox interface has an additional mechanism, as in Burst Read Mode. The INDX portal has special behavior when it has the value 0h through 3h (that is, when INDX is set to point to the Embedded Flash Data Register). Every time the DATA portal is written by the Host, INDX will be automatically incremented by 1 when its value is 0h through 2h. If DATA is written by the Host when INDX is 3h, INDX is set to 0h and the Tail register of the Data FIFO is advanced to the Head and the Tail is marked not Valid. If there is a valid address in the Head of the Address FIFO, the Word Program sequence will be initiated. In Burst Program Mode, the entire contents of the Flash memory can be programmed by the Host with a sequence of writes to DATA, without any intervening writes to INDX.

14.8.4 ERASE MODE

When Embedded Flash Command Register is set to Erase Mode. The state machine will wait until a valid address is written into the Embedded Flash Address Register. Once the register is valid, Busy in the Embedded Flash Status Register is set to '1b and the Embedded Flash controller sequences the Embedded Flash array strobes to erase part or all of the Embedded Flash. All bits in the erased area are set to '1b'. Busy remains set during the operation. When the erase operation is complete Busy is set to '0b'. The controller remains in Erase Mode and can accept additional addresses in the Embedded Flash Address Register.

The area to be erased is determined by Bits[23:18] of the Embedded Flash Address Register, as shown in Table 14-5, "Erase Mode Functions". In all cases Embedded Flash Address Register bits[10:0] are ignored. The Embedded Flash Address Register should be configured with the proper page address before setting the Embedded Flash Command Register to Erase Mode.

TABLE 14-5: ERASE MODE FUNCTIONS

Address Register Bits [23:18]	Area Erased
xxxx00b	2KB Page in Embedded Flash Main Array specified by Embedded Flash Address Register[17:11]
xxxx01b	Embedded Flash Info Block
111110b	Mass erase of entire Embedded Flash Main Array
111111b	Mass erase of entire Embedded Flash Main Array and Embedded Flash Info Block

14.9 Flash Lock Controls

14.9.1 FLASH WRITE PROTECT

When the Boot_Lock in the Embedded Flash Configuration Register is asserted or the input pin nFWP is asserted, the bottom 4K bytes (0000h – 0FFFh) of the Flash Main Memory Array (the Boot Block) are write protected and cannot be changed by any programming method including Program Mode or Erase Mode. The Mass Erase function is therefore disabled, since it would erase the Boot Block. The rest of the Embedded Flash array, including the Info block, is not affected. If both Boot_Lock and the input pin nFWP are not asserted, all programming and erase functions, including Erase Mode, are enabled and the bottom 4K bytes are not protected.

14.9.2 FLASH BOOT PROTECT

The Embedded Flash Boot Block can be protected from all reads and writes, from either the Host or the EC. If Boot_Protect_En in the Embedded Flash Configuration Register is '1b' the first time there is a address reference to the Embedded Flash Subsystem that is outside the Boot Block (that is, the first time the EC does either a program fetch to an address that is larger than 00_0FFFH or a data reference to an address that is larger than 00_0FFFh and less than 80_0000h), Boot_Block and Boot_Lock in the Embedded Flash Status Register are set. Boot_Lock prevents any programming or erase operations on the Boot Block, as described in Section 14.9.1, "Flash Write Protect". Mass Erase is also disabled. In addition, any attempt to read the Boot Block, from either the Instruction Memory Interface or the Register Interface will return FFFF_FFFFh. Once set, only a VTR POR (nSYS_RST) can clear Boot_Block.

Typically, Flash Boot Protect will be the last function performed as part of boot. The EC boot code should configure the Interrupt Vector Table to be outside the Boot Block. It should then set the Boot_Protect_En bit, then jump to a location outside of the Boot Block. From that point on the Boot Block will be protected for both read and write access from either the Host or the EC.

14.9.2.1 JTAG Disable to protect the flash Boot Block

If Boot_JTAG_Block in the Embedded Flash Initialization Register is set to '0b', the JTAG Debug Data Registers interface will not be accessible to the MEC1609/MEC1609i JTAG pins as long as Boot_Block in the Embedded Flash Status Register is '0b'. This insures that an external device cannot read or write the Boot Block while the EC is executing within it. If Boot_JTAG_Block is in its default state of '1b', an external debugger could potentially halt the EC and then read or reprogram any word in the Boot Block.

As described in Section 14.9.4, "Flash Data Initialization", data in the Embedded Flash Initialization Register are configured by programming the initialization vector in the Embedded Flash array.

14.9.3 FLASH DATA PROTECT

The Flash Data Protect Region is the last 4KB in the Embedded Flash main array. If Data_Protect in the Embedded Flash Configuration Register is '1b' the Flash Data Protect Region is protected from any read or write accesses by the Host through the Mailbox interface. The EC can access the Data Protect Region through both the Instruction Memory Interface and the Register Interface, as long as the JTAG port is not enabled. Once JTAG is enabled, by de-asserting the JTAG Reset pin (JTAG_RST#) see Table 2-9, "JTAG Interface," on page 14), Data_Block in the Embedded Flash Status Register is set to '1b' and the entire Data Protect Region is blocked from all accesses, by the Host, by the EC or by JTAG. Any reads return FFFF_FFFFh, and any attempt to program data in the region will be blocked. Any page erase operation in the Data Protect Region is blocked and Mass Erase is also disabled. Once Data_Block is set, it can only be cleared by a VTR POR (nSYS_RST). Clearing Data_Protect in the Embedded Flash Configuration Register will not change Data_Block, so the only way to access the protected region once JTAG is enabled is to power cycle the device.

14.9.4 FLASH DATA INITIALIZATION

The last word in the Embedded Flash Boot Block (the word at the address 0_0FFCh) is used as an initialization vector that is loaded into a read-only register, the Embedded Flash Initialization Register. This register is loaded as soon as VTR POR (nSYS_RST) is de-asserted and before the EC comes out of reset. Once written, this data will stay constant until the upper page of the Boot Block is erased and the VTR power is cycled. The register can be used for configuration and initialization data.

Boot_JTAG_Block of the Embedded Flash Initialization Register is used to control JTAG access during the boot period. If this bit is programmed to '0b', JTAG cannot access any locations within the MEC1609/MEC1609i, other than the JTAG test registers, while the Boot Block is readable. If Boot_Block in the Embedded Flash Status Register is set, JTAG accesses are enabled, although any access to the Boot Block is blocked by Boot_Block. If Boot_JTAG_Block is left in the default '1b' state, a JTAG master could halt the EC just after VTR POR (nSYS_RST) and manipulate all locations in the MEC1609/MEC1609i, including any data in the Boot Block.

14.9.5 EMERGENCY MASS ERASE

If Boot_JTAG_Block of the Embedded Flash Initialization Register is asserted but the EC boot code is faulty, it is possible that the EC could become stuck in the boot sequence without ever enabling JTAG access or LPC access. In that state, there would be no way to erase the Embedded Flash and reprogram the EC, since both the JTAG port and LPC access would be inaccessible. For this reason, a fail-safe mechanism is included in the Embedded Flash controller.

If the ME (Mass Erase) field in the JTAG TEST REGISTER 4/Reset Register (Dh) is '1b' when VTR POR (nSYS_RST) is de-asserted the Embedded Flash controller, after transitioning to the Flash Data Initialization state will transition instead to Erase Mode. Bit[23:19] of the Embedded Flash Address Register will be configured with '11111b', forcing a Mass Erase. After the Mass Erase is completed the Embedded Flash is held in an idle state and the Embedded Controller will remain idle. JTAG and LPC will be blocked as well.

An additional power cycle of the MEC1609/MEC1609i will be required to reset the Embedded Flash Initialization Register and re-enable the Embedded Flash and the Embedded Controller. After the additional power cycle JTAG can be used to reprogram the Embedded Flash.

In order to trigger an Emergency Mass Erase, the JTAG interface should be used to perform the following sequence:

1. Place the JTAG_RST# pin in the asserted (low) and apply VTR Power.
2. Delay 200 ns before deserting the JTAG_RST# pin (high) to take the JTAG interface out of the reset state.
3. The POR EN field in the JTAG TEST REGISTER 4/Reset Register (Dh) should be set to '1b' in order to enable JTAG control of the VTR POR circuitry.
4. The VTR POR field in the JTAG TEST REGISTER 4/Reset Register (Dh) should be set to '1b' in order to generate a VTR POR in the MEC1609/MEC1609i.
5. The ME field in the JTAG TEST REGISTER 4/Reset Register (Dh) should be set to '1b'.

6. The VTR POR field in the JTAG TEST REGISTER 4/Reset Register (Dh) should be set to ‘0b’ which terminates the VTR POR.
7. Delay 50ms to ensure a Mass Erase completes and the VTR POR (nSYS_RST timing) should be set to ‘1b’ in order to generate a VTR POR in the MEC1609/MEC1609i.
8. The VTR POR field in the JTAG TEST REGISTER 4/Reset Register (Dh) should be set to ‘0b’ which terminates the VTR POR.

14.10 Programming the Embedded Flash Array

In normal operation, the EC fetches all of its instructions from the Embedded Flash array. While it is executing out of Flash, the program interface using the registers described in the next section cannot be used. When the Register Interface is used, the EC cannot fetch either instructions or data out of the Embedded Flash. The EC can either sleep, or run entirely out of the EC Data Memory. When running out of EC Data Memory, EC firmware should insure than all interrupts are disabled, unless the interrupt vector table is relocated to the EC Data Memory along with any interrupt handler that is enabled.

There are three methods by which the Embedded Flash array can be programmed:

- The Flash may be programmed through the JTAG interface. Through the JTAG interface, the EC can be halted or directed to run entirely out of the EC Data Memory. Once the EC no longer requires the Flash, the registers in the next section can be used to program the array, by first setting Reg_Ctl_En in the Embedded Flash Configuration Register (enabling register control of the Flash), then using the other registers to program the array. Once programming is completed, Reg_Ctl_En in the Embedded Flash Configuration Register is cleared (enabling EC instruction access) and then setting the EC into the RUN state. See Section 39.0, "JTAG and XNOR," on page 477 for a description of JTAG operation.
- The Flash may be programmed by the Host. After co-ordination between the Host and the EC, the EC configures the Embedded Flash Configuration Register to enable the register programming mode and to give the Host write access to the registers (by setting both Reg_Ctl_En and Host_Ctl to ‘1b’). The EC would then disable all interrupts except the EC_Int Flash Mailbox interrupt from the Host. The EC Interrupts for LRESET# and VCC PWRGD pin signals should also be enabled (see Section 14.4, "Interrupts," on page 222), in order to restore control to the EC in the event the Host loses power and does not return control to the EC properly. The EC then signals the Host that it can proceed. At this point the EC should verify that the Host can take control, by checking the state of LRESET# and VCC PWRGD. If LRESET# is asserted or VCC PWRGD is de-asserted, the Host will not respond to the EC signal to take control, and the EC should therefore cancel the hand off to the Host, clear Reg_Ctl_En and Host_Ctl, re-enable interrupts and continue. If the Host is capable of responding to the EC signal, the EC finally either puts itself to sleep or runs entirely out of the. When the Host finishes programming the Embedded Flash array, it sets EC_Int in the Embedded Flash Command Register to ‘1b’, which wakes the EC.

The Host may also program the Flash through the use of the MailBox Register Interface. The Host communicates the address to be modified and the data to program through a pre-arranged set of mailbox registers, then sends an interrupt to the EC. The EC would then program the Flash while running out of the on-chip SRAM, as described in the following bullet item.

- The EC can itself program the Flash array by loading a program to do so into the on-chip 4KB SRAM. Once the EC is running out of SRAM, it can disable all interrupts and configure the Flash array to be programmable through the registers (by setting Reg_Ctl_En in the Embedded Flash Configuration Register). After programming is completed, the EC can reset the Reg_Ctl bit to make the Flash accessible to instruction fetch.

In all cases, the Embedded Flash Data Register can be updated to set up the next write while the Flash is busy programming the current write.

APPLICATION NOTE: When the ARC is put to sleep during Flash programming, the ROSC_SLP_OVRD bit in the Clock Control Register on page 101 should be cleared to ‘0’ insure that the ring oscillator never stops.

14.10.1 TRANSFERRING CONTROL TO THE HOST

In order to transfer control of the Embedded Flash to the Host, the EC has to enable the Host access to the Register interface, inform the Host that it has access, then either sleep or execute code entirely out of the EC Data Memory. The EC must sleep or execute in RAM because it will no longer be able to use the ICCM instruction memory interface. The following sequence is an example of how this transfer can be accomplished:

```
// This code is located in the Flash
// Command_Address is 0xFF3908
// Config_Address is 0xFF3910
//
*Config_Address |= REG_CTL_EN | HOST_CTL; // Permit the Host to control the Flash
registers
Disable_all_interrupts();
Enable_Interrupt(EC_Int);         // Enable transition from Host back to EC
Enable_Interrupt(LRESET#);
Enable_Interrupt(VCC_PWRGD);
Host_Message(FLASH_READY);        // Tell Host it can take control of the Flash
if( assserted(LRESET) || deasserted(VCCPWRGD) ) {
    cancel_flash_transfer();      // Host is not powered: annul transfer
                                  // restore Flash configuration/command registers
    restore_interupts();
    return();                     // return to regular EC code
} else {
    sleep();                      // Wait for EC_Int
}
```

The next code sequence would be executed by the Host once it receives the message from the EC that it may take control of the Flash:

```
// INDX is the address of INDX in the Host I/O space
// DATA is the address of DATA in the Host I/O space
//
INDX = 9; // index of Command register byte where Reg_Ctl bit is located
DATA = REG_CTL; Take control of the Flash Address and Data regsiters
```

As an alternative to the Flash-based EC code described above, the transfer function can run partially in Flash and partially in SRAM. In this mode it is not necessary to enable LRESET# or VCC PWRGD interrupts, since SRAM-based EC code can check the state of those signals through polling: Normal termination of Flash programming occurs as before when the Host sets the EC_Int bit in the Command register.

```
// This code is located in the Flash
// Command_Address is 0xFF3908
// Config_Address is 0xFF3910
//
*Config_Address |= REG_CTL_EN;    // Permit control of Flash via register interface
Disable_all_interrupts();
Enable_Interrupt(EC_Int);         // Enable transition back to EC
goto SRAM_Flash_Code;
...
SRAM_Flash_Code:                  // start of code in SRAM
*Command_Address |= REG_CTL;
*Config_Address |= HOST_CTL;
Host_Message(FLASH_READY);
while(true) {                     // commence polling on LRESET/VCC PWRGD
    if( asserted(LRESET) || deasserted(VCCPWRGD) ) {
          cancel_flash_transfer();// Host is not powered: annul transfer
                                  // restore Flash configuration/command registers
          restore_interrupts();
          return();               // return to EC code in Flash
    }
}
```

14.10.2 READING THE EMBEDDED FLASH

The Embedded Flash memory, in both the main array as well as the Info block, can most conveniently be read with Burst mode set. An initial Flash address should be configured, and then the rest of the Flash memory can be read with just a sequence of reads to the data register. The following pseudocode illustrates how reading the Flash from the Host might proceed:

```
// Byte_array[] is a data structure to receive the data. It is a sequence of bytes
// Flashbase is the first address in the flash memory to be read
// Limit is the total number of bytes to read
// INDX is the address of INDX in the Host I/O space
// DATA is the address of DATA in the Host I/O space
//
// this code works as long as the read does not wrap around the end of the Flash array
//
INDX = 8; // index of Command register
DATA = READ | BURST;
INDX = 6; // bits 24:16 of Flash Address register
DATA = Flashbase.byte2;
INDX = 5; // bits 15:8 of Flash Address register
DATA = Flashbase.byte1;
INDX = 4; // bits 7:0 of Flash Address register
DATA = Flashbase.byte0;      // the Flash address advances in the Address FIFO
                             // Read from the Flash is started
INDX = 0; // bits 7:0 of Flash Data register
for( i = 0; i < Limit; i++)
{
    Byte_array[i] = DATA;
}
INDX = 8; // index of Command register
DATA = STANDBY; // Exit READ mode and flush Address/Data FIFOs
```

Reading from the EC is similar, except that data can be read four bytes at a time:

```
// Word_array[] is a data structure to receive the data.
//   It is a sequence of 32-bit words
// Flashbase is the first address in the flash memory to be read
// Limit is the total number of 32-bit words to read
// Flash_Address is 0xFF3904
// Data_Address is 0xFF3900
// Command_Address is 0xFF3908
//
// this code works as long as the read does not wrap around the end of the Flash array
//
*Command_Address = READ | BURST;
*Flash_Address = Flashbase;      // the Flash address advances in the Address FIFO
for( i = 0; i < Limit; i++)
{
    Word_array[i] = *Flash_Data;
}
*Command_Address = STANDBY; // Exit READ mode and flush Address/Data FIFOs
```

14.10.3 WRITING THE EMBEDDED FLASH

The Embedded Flash memory, in both the main array as well as the Info block, can most conveniently be programmed with Burst mode set. An initial Flash address should be configured, and then the rest of the Flash memory can be written with just a sequence of writes to the data register. Because programming takes between 20 and 40 µs, the Host may choose to approximate the latency before polling the Status register. The following pseudocode illustrates how reading the Flash from the Host might proceed:

```
// Byte_array[] is a data structure that sources the data. It is a sequence of bytes
// Flashbase is the first address in the flash memory to be programmed
// Limit is the total number of bytes to write. It is assumed to be a multiple of
//    the 256-byte row size for simplicity
// INDX is the address of INDX in the Host I/O space
// DATA is the address of DATA in the Host I/O space
//
// this code works as long as the write block does not wrap around
// the end of the Flash array.
//
INDX = 8; // index of Command register
DATA = PROGRAM | BURST;
INDX = 6; // bits 24:16 of Flash Address register
DATA = Flashbase.byte2;
INDX = 5; // bits 15:8 of Flash Address register
DATA = Flashbase.byte1;
INDX = 4; // bits 7:0 of Flash Address register
DATA = Flashbase.byte0;      // the Flash address advances in the Address FIFO
for( i = 0; i < Limit; )
{
    INDX = 0xC; // index of Status register, to check space in Data FIFO
    while( DATA & DATA_FULL ); // stall while Data FIFO has no space
    INDX = 0; // bits 7:0 of Flash Data register
    // Write one word (four bytes) into the Flash array
    DATA = Byte_array[i++];
    DATA = Byte_array[i++];
    DATA = Byte_array[i++];
    DATA = Byte_array[i++];
}
INDX = 8; // index of Command register
DATA = STANDBY; // Exit PROGRAM mode and flush Address/Data FIFOs
```

Writing from the EC is similar, except that data can be written four bytes at a time:

```
// Word_array[] is a data structure to receive the data.
//   It is a sequence of 32-bit words
// Flashbase is the first address in the flash memory to be written
// Limit is the total number of 32-bit words to write
// Flash_Address is 0xFF3904
// Data_Address is 0xFF3900
// Command_Address is 0xFF3908
// Status_Address is 0xFF390C
//
// this code works as long as the program block does not wrap around
// the end of the Flash array
//
*Command_Address = PROGRAM | BURST;
*Flash_Address = Flashbase;      // the Flash address advances in the Address FIFO
for( i = 0; i < Limit; i++)
{
    while( *Status_Address & DATA_FULL ); // stall while Data FIFO has no space
    *Flash_Data = Word_array[i];
}
*Command_Address = STANDBY; // Exit PROGRAM mode and flush Address/Data FIFOs
```

14.10.4 ATE PROGRAMMING

There is no ATE-specific interface for programming the Embedded Flash. ATE will use either the LPC register interface, or the EC interface via JTAG.

14.11 Detailed Description of Accessible Registers

See Section 14.7, "Register Interface," on page 223.

14.11.1 EMBEDDED FLASH DATA REGISTER

The Embedded Flash Data Register can only be written when the Embedded Flash Controller is in Program Mode. In Standby Mode, Read Mode and Program Mode the register is read-only. Writes will complete but have no effect.

TABLE 14-6: EMBEDDED FLASH DATA REGISTER

HOST INDEX	BYTE3: INDX 03h BYTE2: INDX 02h BYTE1: INDX 01h BYTE0: INDX 00h			8-bit		**HOST SIZE**		
EC OFFSET	100h			32/16/8-bit		**EC SIZE**		
POWER	VTR			0000_0000h		nSYS_RST **DEFAULT**		
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[31:24]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[7:0]							

FLASH_DATA[31:0]

This 32-bit register holds the data to be written into the Flash memory array during a program cycle, as well as the data returned from a Flash memory read during a Read From SPB cycle. It should be set up before the Embedded Flash Address Register is configured.

14.11.2 EMBEDDED FLASH ADDRESS REGISTER

TABLE 14-7: EMBEDDED FLASH ADDRESS REGISTER

HOST INDEX	BYTE3: INDX 07h BYTE2: INDX 06h BYTE1: INDX 05h BYTE0: INDX 04h			8-bit			**HOST SIZE**	
EC OFFSET	104h			32/16/8-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Address[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Flash_Address[7:2]						Reserved	

FLASH_ADDRESS[23:0]

This register represents a byte address for the Embedded Flash array. Since all read and write operations are to 32-bit quantities, the low-order to bits (Flash_Address[1:0] must always be 0, so these bits are reserved. If the Flash state machine is in Read Mode, writing Byte0 of the Embedded Flash Address Register initiates the Read sequence. If the Flash state machine is in Program Mode, the Program sequence is initiated when there is both a valid Data value in the Embedded Flash Data Register and the Embedded Flash Address Register has been updated by writing Byte0. If Bit18 is ‘1b’ then the Info block in the Embedded Flash subsystem is selected. Bit19 is only examined in Erase Mode. When in Erase Mode, if Bits[23:19] contain any value other than ‘11111b’, then Page Erase is enabled. If Bits[23:19] is ‘11111b’, Mass Erase is enabled.

14.11.3 EMBEDDED FLASH COMMAND REGISTER

This register is Read-Only while Busy in the Embedded Flash Status Register is ‘1b’. An attempt to write this register while Busy is asserted will not modify the register and will set Busy_Err in the Embedded Flash Status Register.

A write to this register causes the Embedded Flash controller to transition to the selected state.

TABLE 14-8: EMBEDDED FLASH COMMAND REGISTER

HOST INDEX	BYTE0: INDX 8h BYTE1: INDX 9h			8-bit		**HOST SIZE**		
EC OFFSET	108h			32/16/8-bit		**EC SIZE**		
POWER	VTR			00h		nSYS_RST **DEFAULT**		
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Reg_ Ctl
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
EC TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EC_ Int	Burst	Flash_Mode[1:0]	

FLASH_MODE

The field determines which Master may write the registers in the Embedded Flash controller:

0 The Embedded Flash controller is placed in the Standby mode.

1 The Embedded Flash controller is placed in Read mode.

2 The Embedded Flash controller is placed in Program mode.

3 The Embedded Flash controller is placed in Erase mode.

BURST

If the Embedded Flash controller is in Read Mode or Program Mode and this bit is ‘1b’, the contents of the Head register in the Embedded Flash Address Register FiFO will be incremented by 4 and written into the Tail register whenever the Head register is Valid. When this bit is ‘0b’, the Embedded Flash Address Register and the INDEX register are not incremented automatically.

See Section 14.8.2.2, "Burst Read Mode," on page 227 and Section 14.8.3.2, "Burst Program Mode," on page 228 for information about the use of Burst.

EC_INT

Setting this bit to ‘1b’ generates an EC interrupt and simultaneously sets Reg_Ctl_En and Host_Ctl in the Embedded Flash Configuration Register as well as Reg_Ctl_En in this register to ‘0b’. By setting these bits to ‘0b’ the Embedded Flash array is configured for EC instruction access, so that when the EC awakes and responds to the interrupt request triggered by EC_Int, it can fetch the interrupt vector from its usual location in the Embedded Flash array. See Section 14.4, "Interrupts," on page 222.

The response to setting this bit is the same as the response to VCC PWRGD or PCIRESET signals, as detailed in Table 14-1, "VCC PWRGD and LRESET# Behavior". If the Embedded Flash Controller is in either Erase Mode or Program Mode the programming sequence is terminated, the Embedded Flash controller is set to Standby Mode, Reg_Ctl, Reg_Ctl_En and Host_Ctl are set to ‘0b’ and the wakeup interrupt is sent to the EC.

REG_CTL

When this bit is set, the address input and the control strobes of the Flash Memory Array are sourced from the registers in the Embedded Flash Subsystem. Software on either the Host or the EC can read and write the Flash Memory Array by reading and writing the registers in the subsystem. The EC cannot execute instructions out of the Flash Memory Array.

When this bit is cleared (which is the default), the address input and the control strobes of the Flash Memory Array are sourced from the Instruction Closely Coupled Memory interface. The EC can execute instructions directly from the Flash Memory Array. The other registers in the Embedded Flash Subsystem can be read or written, but do not affect the Flash Memory Array.

Note: A 32-bit write from either the EC or JTAG with Reg_Ctl_En set to ‘1b’ can simultaneously write the Byte 0 control bits in the Embedded Flash Command Register.

This bit can only be set if Reg_Ctl_En in the Embedded Flash Configuration Register is ‘1b’.

14.11.4 EMBEDDED FLASH STATUS REGISTER

TABLE 14-9: EMBEDDED FLASH STATUS REGISTER

HOST INDEX	BYTE 1: INDX Dh BYTE 0: INDX Ch			8-bit			**HOST SIZE**	
EC OFFSET	10Ch			32/16/8-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R	R	R	R	R	R/WC	R/WC	R/WC
EC TYPE	R	R	R	R	R	R/WC	R/WC	R/WC
BIT NAME	Reserved					Protect_Err	CMD_Err	Busy_Err
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved	Data_Block	Boot_Block	FWP	Boot_Lock	Address_Full	Data_Full	Busy

BUSY

This bit reflects the state of the Embedded Flash controller. This bit is set while the controller is processing a flash control sequence. While this bit is set the Embedded Flash Command Register can not be written. If a write is attempted on this register, the write fails and Busy_Err in this register is set.

This bit is read-only.

DATA_FULL

This bit reflects the value of the Valid bit of the Tail register of the Embedded Flash Data Register. When it is set the FIFO is full and any additional writes will set Busy_Err.

This bit is read-only.

ADDRESS_FULL

This bit reflects the value of the Valid bit of the Tail register of the Embedded Flash Address Register. When it is set the FIFO is full and any additional writes will set Busy_Err.

This bit is read-only.

BOOT_LOCK

This bit is set whenever the Boot Block is write-protected. The Boot Block will be write-protected when Boot_Lock in the Embedded Flash Configuration Register is set, or when the nFWP input pin is asserted.This bit is also set when Boot_-Block is set, so that whenever the Boot Block is read-protected it is also write-protected.

A copy is maintained in this register so that the Host can access it, since the Host has no access to the Embedded Flash Configuration Register. See Section 14.9.1, "Flash Write Protect," on page 229 for a description of write-protecting the boot block.

This bit is read-only.

FWP

The FWP bit reflects the state of the nFWP input pin. When the nFWP pin is asserted (‘0b’), the FWP bit is ‘1b’. When the nFWP pin is not asserted (‘1b’), the FWP bit is ‘0b’. See Section 14.9.1, "Flash Write Protect," on page 229 for a description of write-protecting the boot block.

BOOT_BLOCK

When this bit is ‘1b’, the Boot Block is protected from all access. See Section 14.9.2, "Flash Boot Protect," on page 229. Once set, it will only be cleared by a VTR Power On Reset.

This bit is read-only.

DATA_BLOCK

When this bit is ‘1b’, the Protected Data Block is protected from all access. See Section 14.9.3, "Flash Data Protect," on page 230. Once set, it will only be cleared by a VTR Power On Reset.

This bit is read-only.

BUSY_ERR

This bit is set if

- A write to the Embedded Flash Command Register occurs while Busy is set.
- A write to the Embedded Flash Address Register occurs while Address_Full is set.
- A write to the Embedded Flash Data Register occurs while Data_Full is set

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

CMD_ERR

If the Embedded Flash controller is in Read Mode, Program Mode or Erase Mode, this bit is set if the Embedded Flash Command Register is set to any value other than Standby Mode.

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

PROTECT_ERR

If Boot_Lockor Boot_Block in the Embedded Flash Status Register is set and a program operation to an address within the range 0000h - 0FFFh, or a page erase operation to page 0, or a mass erase operation to the main Embedded Flash array occurs, this bit is set. In addition, if Boot_Block is set and any read access to the Embedded Flash in the range 0000h through 0FFFh is attempted, either through the Instruction Memory Interface or the Register Interface, this bit will be set.

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

14.11.5 EMBEDDED FLASH CONFIGURATION REGISTER

TABLE 14-10: EMBEDDED FLASH CONFIGURATION REGISTER

HOST INDEX	N/A			N/A			**HOST SIZE**	
EC OFFSET	110h			32/16/8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		INHIBIT_ JTAG	Data_ Protect	Boot_ Protect_E n	Boot_ Lock	Host_ Ctl	Reg_ Ctl_En

Note: This register is accessible only by the EC.

REG_CTL_EN

When this bit is set, Reg_Ctl of the Embedded Flash Command Register can be set to 1. When this bit is clear (the default), Reg_Ctl is forced to 0 and cannot be set.

Because this bit overrides any writes to Reg_Ctl, EC firmware can clear this bit to prevent the Host from getting access to the Embedded Flash register interface.

HOST_CTL

When this bit is set, the Embedded Flash Address Register, the Embedded Flash Data Register and the Embedded Flash Command Register can be read and written by the Host via the Flash Memory Mailbox registers in the Flash Mailbox interface. The EC is inhibited from reading or writing these registers. Writes have no effect and all reads return all 0’s.

When this bit is clear (the default), these registers can be read and written by the EC. The Host is inhibited from reading or writing these registers. Writes have no effect and all reads return all 0’s.

The Embedded Flash Status Register is always readable by both the Host and the EC, independent of the state of Host_Ctl.

BOOT_LOCK

The Boot_Lock bit permits the EC to lock the Flash boot block when the nFWP input pin is not asserted (see Table 2-15, “Miscellaneous Functions,” on page 17). When Boot_Lock is ‘1b’, the Flash boot block is locked regardless of the state of the nFWP input pin. Any attempt to write data in the address range 00000h through 00FFFh in the Embedded Flash address space will fail and set Busy in the Embedded Flash Status Register. When Boot_Lock is ‘0b’ (the default), the Flash boot block is unlocked if the nFWP pin is not asserted.

BOOT_PROTECT_EN

When this bit is set, the Boot Protect function is enabled. The first time the EC does either a program fetch to an address that is larger than 00_0FFFH or a data reference to an address that is larger than 00_0FFFh and less than 80_0000h causes Boot_Block in the Embedded Flash Status Register to be set and all further access to the Boot Block will be prohibited. See Section 14.9.2, "Flash Boot Protect," on page 229.

DATA_PROTECT

If this bit is ‘1b’, the top two 2KB pages in the Embedded Flash array are not readable or writable through the Host access mailbox. The two pages are accessible by the EC through both the Instruction Memory Interface and the Register Interface, as long as the JTAG port is not enabled. Once JTAG is enabled on the pins, Data_Block in the Embedded Flash Status Register is set and the two pages become inaccessible to all interfaces.

If this bit is ‘0b’ (the default), the top two pages can be read and written normally.

INHIBIT_JTAG

When this bit is ‘1b’, the JTAG interface is blocked from any access to the EC or the internal buses in the MEC1609/MEC1609i. Only the registers in the JTAG interface are accessible. If this bit is ‘0b’, accesses by the JTAG interface is blocked if is Boot_JTAG_Block n the Embedded Flash Initialization Register is ‘0’ and Boot_Block in the Embedded Flash Status Register is ‘0’.

14.11.6 EMBEDDED FLASH INITIALIZATION REGISTER

The Embedded Flash Initialization Register is a read-only register that is loaded from address 0_0FFCh in the Embedded Flash Array when VTR Power On Reset is de-asserted but while the EC is still held in Reset. The address is the last 32-bit word in the Boot Block.

This register is reset by hardware to be all 0’s. The bit Boot_JTAG_Block will be 0, which blocks JTAG access to the ARC address space. If the read of the Flash is successful, the value of this register will be FFFF_FFFFh when the Flash is fully erased. In this state JTAG is not blocked.

TABLE 14-11: EMBEDDED FLASH INITIALIZATION REGISTER

HOST INDEX	N/A			N/A			**HOST SIZE**	
EC OFFSET	114h			32/16/8-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[31:24]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[7:1]							Boot_ JTAG_ Block

Note: This register is accessible only by the EC.

BOOT_JTAG_BLOCK

If this bit is ‘0b’ the JTAG interface is blocked from any access to the MEC1609/MEC1609i as long as Boot_Block in the Embedded Flash Status Register is ‘0b’. This means that as long as the Boot Block is accessible by the EC an external JTAG function cannot read or write any address inside the device, including the Boot Block. Once boot code renders the Boot Block inaccessible by causing Boot_Block to be set, JTAG functionality is enabled. JTAG cannot read or write any data in the Embedded Flash Boot Block, including the data used to load the Embedded Flash Initialization Register.

If this bit is ‘1b’ (the value that will be loaded if the Boot Block is erased), JTAG access will be enabled as soon as VTR POR is de-asserted.

BITS[31:1] INITIAL_DATA

The data in this field are loaded after VTR POR is de-asserted, along with Boot_JTAG_Block. The data can be used for device identification or configuration.

14.11.7 FLASH MAILBOX INDEX REGISTER

TABLE 14-12: FL_MBX_INDEX REGISTER

HOST OFFSET	00h			8-Bit			**HOST SIZE**	
EC OFFSET	00h			8-Bit			**EC SIZE**	
POWER	VTR			00h			**VCC POR DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	INDEX7	INDEX6	INDEX5	INDEX4	INDEX3	INDEX2	INDEX1	INDEX0

14.11.8 FLASH MAILBOX DATA REGISTER

TABLE 14-13: FL_MBX_DATA REGISTER

HOST OFFSET	04h			8-Bit			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VTR			00h			**VCC POR DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

15.0 ARC 625D EMBEDDED CONTROLLER

15.1 General Description

This chapter contains a description of the Embedded Controller used in the MEC1609/MEC1609i.

The Embedded Controller on the MEC1609/MEC1609i is an ARC 625D Processor by ARC International. The ARC625D is a full-featured 32-bit embedded processor. Its features include:

- 5-stage instruction pipeline with single-cycle instruction execution
- Static branch prediction
- 32-bit data, instruction and address buses
- 16- and 32-bit instructions, with no overhead for switching between 16- and 32-bits
- 32 32-bit general purpose registers
- Scoreboarded data memory pipeline to reduce data stalls
- Debug features
  |Debug host can access all registers and CPU memory, with a JTAG interface to host tools
  Multiple action points for real-time instruction and data breakpoints
- Industry standard AHB system interface
- Power saving features
  Sleep mode via software instruction
  Clock gating
- Two highly configurable action points for debugging

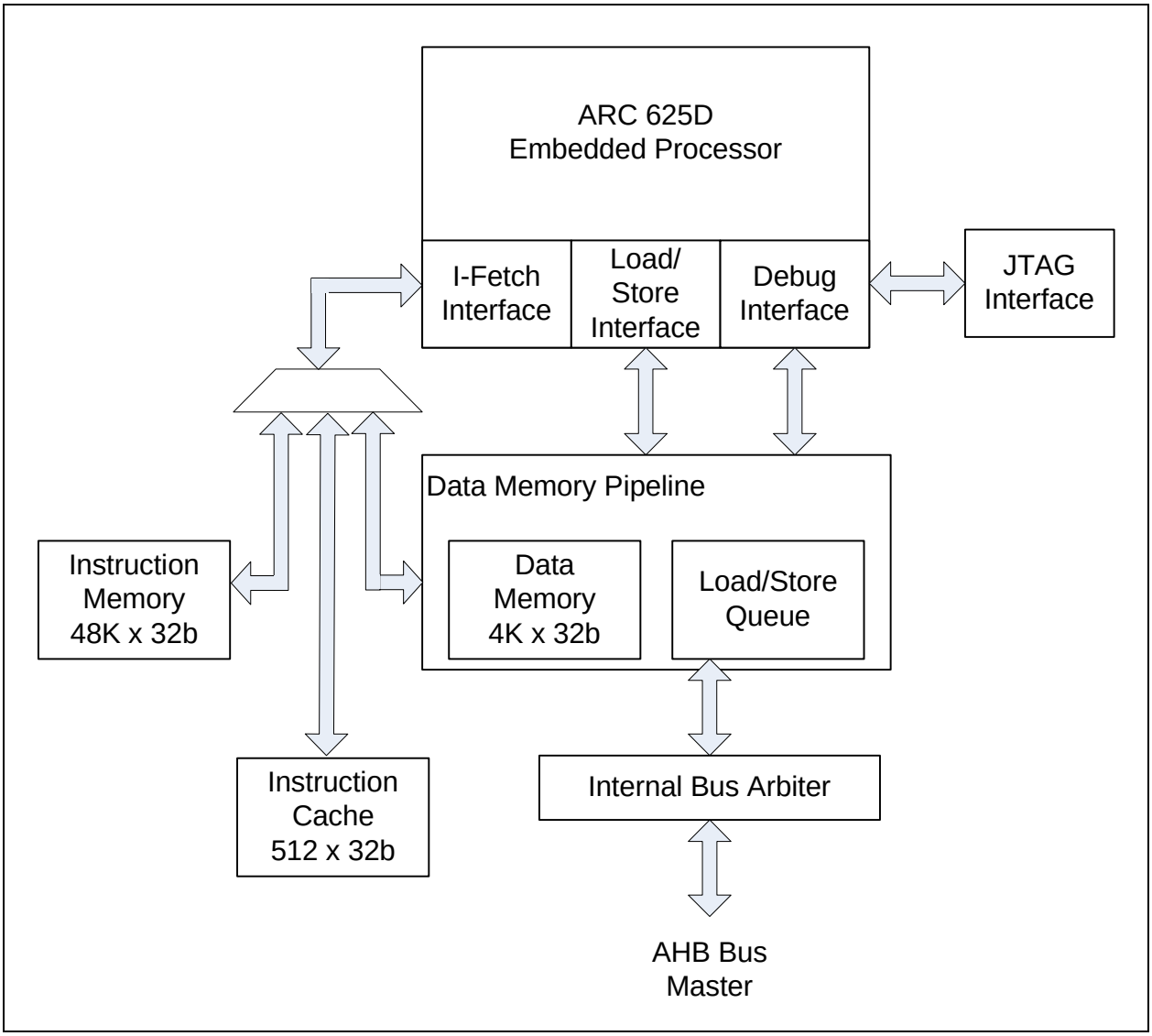
The ARC625D is highly configurable. The configuration used in the MEC1609/MEC1609i incorporates:

- 192KB plus 2KB single cycle Embedded Flash Closely Coupled instruction memory
- 16-KB Single Cycle 32-bit wide dual-ported SRAM, accessible as both Closely Coupled Data Memory and Instruction Memory
- 2-KB direct mapped instruction cache
- Interrupt controller with 32 interrupts
- Normalize instruction, which can find leading ones and zeros in a word
- Multiply instruction, which completes a 32x32 multiply in 3 cycles
- Divide Assist instruction
- Two full-featured Actionpoints (Dedicated Breakpoint Blocks), which can trigger breakpoints on both instruction accesses and data access

For details on the architecture of the ARC625D processor, see ARC International's *ARCompact™ Instruction Set Architecture Programmer's Reference*, March 2005.

15.2 Block Diagram

FIGURE 15-1: ARC BLOCK DIAGRAM



15.3 EC Clocking

The ARC 625D Embedded Controller can be configured to run at clock at rates ranging from 4.3 MHz to 32 MHz. Operating frequencies higher than 21.5 MHz can only be used to execute code out of the SRAM, and only when this code does not contain misaligned 32-bit instructions or long intermediate operands, which can be ensured by turning on appropriate compiler switches (at the expense of larger memory footprint). EC clock rates are set via the EC Clock Divider Register on page 102. See Section 5.0, "Power, Clocks and Resets," on page 73.

Table 15-1 describes the restrictions related to code execution from SRAM with respect to code alignment; Table 15-2 describes the restrictions related to code execution from the Embedded Flash Subsystem with respect to EC Clocking. Note that there are no code alignment restrictions when the ARC 625D Embedded Controller is running from the Embedded Flash Subsystem.

TABLE 15-1: CODE EXECUTION FROM SRAM

EC Clock Divider Register	32 Bit Aligned Code and No Long Intermediate Operands?	Description
X	NO	Not allowed
	YES	Allowed

Note: In Table 15-2 X = Don't Care.

TABLE 15-2: CODE EXECUTION FROM EMBEDDED FLASH

EC Clock Divider Register	32 Bit Aligned Code and No Long Intermediate Operands?	Description
2	X	Not allowed
>2		Allowed

Sleep mode via software instruction initiates power saving via Clock gating and ultimately by stopping 64.52 MHz Ring Oscillator. See Section 5.4.7, "Power Management Interface," on page 83 for detailed description of Block Clocking Model and power management.

15.4 EC Memory Map

The ARC processor executes code out of the EC Instruction Memory. This instruction memory is an ARC ICCM (Instruction Closely-Coupled Memory) and each 32-bit word can be accessed in one processor cycle. Data references can come from either the EC Data Memory or from addresses located in the AHB Address Space. The EC Data Memory is an ARC DCCM (Data Closely-Coupled Memory), so each 32-bit word can read or written in one cycle.

See ARC Address Space on page 46 for further details on the ARC address space.

15.4.1 EC DATA MEMORY

The EC has a 16KB Closely Coupled Data memory, implemented with static RAM and organized 4K x 32 bits. Loads and stores to this memory are completed in one cycle. The base address of the memory is 80_0000h in the EC address space and extends to location 80_3FFFh. The EC cannot execute instructions from this address range.

The 16KB Data memory also appears in the instruction space in the address range 6_0000h through 6_3FFF as described in Section 15.4.2, "EC Instruction Memory".

15.4.2 EC INSTRUCTION MEMORY

The primary instruction memory for the EC is a 48K x 32 bit Embedded Flash memory, located at locations 00_0000h through 02_FFFFh in the EC address space. There is also an additional 512 x 32b block of Flash memory, called the Info block, located at addresses 04_0000h through 04_07FFh. Instruction fetches to these two blocks complete in one cycle. The ARC can access locations in the Embedded Flash Memory through load and store instructions, with one additional processor cycle penalty.

The region 03_0000h - 03_FFFFh is not populated and should be accessed - the result would be alias to 01_0000h - 01_FFFFh.

The 16KB Data memory also appears in the instruction space in the address range 6_0000h through 6_3FFF. The memory is dual-ported, so instruction fetches from this space can occur in parallel with data loads and stores, without wait states for either instruction fetch or data reference. If the Embedded Flash memory is configured to be only accessible via the register interface (see Section 15.8, "EC Registers," on page 249), the EC can execute instructions out of the SRAM. For example, the EC could run code that programs the Flash while the Embedded Flash memory is set for the register interface.

RAM_Select in the AHB SRAM Configuration Register can be used to disable instruction access to the SRAM. If instruction access to the SRAM is not needed, disabling it saves power.

Instruction fetches in the range of 00_0000h through 7F_FFFFh do not incur bus errors. Any instruction fetch to non-existent memory will return FFFF_FFFFh.

15.4.3 EC INSTRUCTION CACHE

A 2-kB direct mapped instruction cache, organized as 512 x 32 with 16-byte line size, is provided. The cacheable memory range is 40_0000h to 7F_FFFFh. On cache read misses the ARC core issues fetch requests to its AHB master interface. If enabled, such requests will be serviced by the SPI Flash controller, retrieving missed instructions from the external SPI memory.

Software can optionally allocate half of the cache to store contents of memory while the other half serves as direct mapped cache. This is functionally equivalent to locking half of the cache while still making the whole (4 MB) memory cacheable. Please refer to the ARC600 Cache Reference document for details on cache-related registers.

15.5 ARC Pipelining

The ARC625D processor is pipelined with five pipe stages. Loads and stores are further pipelined through the Load/Store Queue as shown in Figure 17.1, “ARC Block Diagram”, so loads and stores will take additional cycles to complete. The AHB bus is also pipelined. Because of the different pipelines, it is difficult to determine exactly how long a load or store to a register will take if the register is located on either the LPC SPB bus or the EC SPB bus.

Because the ARC processor issues all instructions in order and resolves data hazards within the pipeline, software will typically not have to consider pipeline effects. Stores will complete in the order issued and no load instruction will return data until all stores issued previously have completed. However, there may be some situations in which it is necessary to ensure that pipelines have flushed and all stores have completed before further code execution. The following three instruction sequence provides this outcome:

1. STORE to a memory location
2. LOAD from the same memory location to a processor register
3. Issue any instruction that uses the register in step 2) as one of its sources

The following assembly code is an example of the sequence:

```
; R0 = a value to be written to an AHB memory location
; R1 = the AHB address of the location to be written
;
ST R0, [R1,0]            ;; store
LD R0, [R1,0]             ; load from same location
ADD R0, R0, 0 ; dummy instruction dependent on R0
```

15.6 EC AHB Bus Interface

The ARC Embedded Controller has a single AHB Bus Master interface; see Section 3.3.2, "AHB Address Space," on page 48. The ARC can have at most one access pending on the AHB at one time. The ARC can perform 8-bit, 16-bit and 32-bit loads and stores on the AHB. Instruction fetches over the AHB can take the form of either a 32-bit word load or a 4-word (128-bit) cache line fill.

Possible AHB bus errors are described in Section 3.4.3, "AHB Bus Errors," on page 52. The ARC processor responds to a bus error with Memory Error exception. The first address that caused a memory error is recorded in the AHB Error Address Register. Because ARC exceptions are imprecise, and since several bus errors can occur between the time a bus error address is recorded and the time the ARC processes the exception, it is not always possible to determine which instruction caused the bus error.

15.7 Actionpoints (Dedicated Breakpoint Blocks)

Actionpoints are defined in the ARC 600 Ancillary Components Manual, Chapter 4. They are dedicated hardware blocks that provide an alternative source of breakpoints when the debugger cannot write to memory (e.g., the code being debugged is in ROM). They also provide the ability to break on memory or Aux register accesses.

The primary justification for including Actionpoints in the design is to provide breakpointing for code in ROM, while code is running at full speed (as opposed to being single-stepped). The debugger by default prefers to write Breakpoint instructions (BRK_S: 7FFFh) into memory in order to perform breakpoints at specified PC values. It will instead use actionpoints if:

- The memory area is declared as ROM,
  - or -
- The flag “-off=prefer_soft_bp“ is given to the debugger.

Actionpoints are controlled by a dedicated set of Aux Registers, in the range 220h - 237h, organized as 3 registers per actionpoint. These are:

- AMV: A 32-bit value (Address or sometimes Data). This register supplies the initial trigger value, as masked by the AMM register. Upon triggering, it is over-written by hardware with the exact value seen.
- AMM: A 32-bit mask applied to the AMV register, making any desired bits don't-cares.
- AC: Control register, selecting modes

The status of all actionpoints is visible in the Aux Register DEBUG, at 5h.

The MEC1609/MEC1609i incorporates two full-featured ARC Actionpoints, Actionpoint 0 and Actionpoint 1.

15.7.1 ACTIONPOINT CONFIGURATIONS

Actionpoints may configured (at processor HW build) to be one of two configurations, Minimal or Full.

Minimal actionpoints may be configured to do the following:

- Trigger on an access by address and access type (Instruction, Bus access, or Aux Register)
- Instruction breakpoints trigger on execution at the address, not at the fetch itself
- Act by either Halt (debugger acts) or SW Interrupt (target SW acts)
- Qualify between Reads and Writes (or both)
- Qualify by masking bits of the address
- Invert Condition (Trigger if No Match)
- Gang actionpoints in pairs or quads: both/all must match

Full actionpoints add the following capabilities:

- Match on opcode for instruction fetches
- Match on data value for data read/write in Aux registers or memory
- Two 34-bit inputs from arbitrary sources (per actionpoint)

15.7.2 SIGNIFICANT LIMITATIONS

Address ranges may degrade performance. Because address matching is bit-masked, it may take multiple actionpoints to refine an address range. Even then, the final range is liable to be too big. The debugger allows a range to be too big, and continues from the breakpoint if the resulting trigger was not in the desired range. Note that this means that the program was being halted at undesired / unexpected times, and so is not running at full speed. A reliable way to avoid this is to specify a range only as a power of 2 in size, aligned on a boundary that is also a power of 2, the same or larger than the range.

There is no way to trigger on both the value and the address of a bus read (Memory, I/O), because the data and the address are not present simultaneously. An Aux register access, however, can trigger on data when either read or written. Do not try to enable Read and Write in the same actionpoint, because that will select only the Write data bus to monitor.

15.7.3 DEBUGGER SUPPORT

As of version 7.4, the debugger supports:

1. Break on Instruction fetch by address

   Actionpoint is used if ROM detected or "-off=prefer_soft_bp" argument is specified
2. Break on Memory Space data accesses

   Read/Write or Both

   Address and Mask

   Range, if size is power of two and target aligned to a power of two (requires 2 Minimal actionpoints, paired)

   Value and Mask (requires 2 Minimal actionpoints, paired)

   Value and Range (requires 4 Full actionpoints, quadded)
3. Aux Register accesses

   Read/Write or Both

15.8 EC Registers

TABLE 15-3: AHB SRAM CONFIGURATION REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	F0_FC00h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3-1] BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							RAM_ Select

RAM_SELECT

When this bit is clear (the default case), the 16KB on-chip SRAM that is part of the EC can only be accessed by loads and stores starting at address 80_0000h. The EC can read and write data in the SRAM at addresses starting at 80_0000h but cannot directly execute instructions.

When this bit is set, the 16KB SRAM is configured to be simultaneously accessible in the address range address 6_0000h through 6_3FFFh. The EC can execute directly out of the SRAM. The EC can still read and write data in the SRAM, with no time penalty per load or store.

TABLE 15-4: AHB ERROR ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	F0_FC04h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_Addr[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_Addr[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_Addr[7:0]							

EC_ADDR[23:0]

If an AHB bus error occurs as the result of an EC AHB bus access, the address that caused the error is held. Once an address is held, additional bus errors are ignored, so this register records the first AHB address that caused an AHB bus error. Any write to this register re-enables capturing AHB bus addresses.

16.0 EC INTERRUPT AGGREGATOR

16.1 General Description

The EC Interrupt Aggregator works in conjunction with the ARC625D processor's interrupt interface to handle hardware interrupts and exceptions.

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

Interrupts are typically asynchronous and are maskable. As shown in Figure 16-1, certain interrupts are connected to the processor, but the majority are connected to the EC Interrupt Aggregator. The latter latches, arbitrates, and forwards the highest-level active interrupt to the processor's IRQ3 interrupt input. It also generates a jump vector associated with the selected interrupt. This vector is made available in one of the processor core's registers and is used to address a location in the Interrupt Vector Table (in memory) that contains the address of the interrupt handler.

The aggregator provides four priority levels for incoming interrupts. The processor provides three: mid- and low priorities for interrupts and high priority for exceptions.

Interrupts classified as wake events can be recognized without a running clock, e.g., while the MEC1609/MEC1609i is in sleep state.

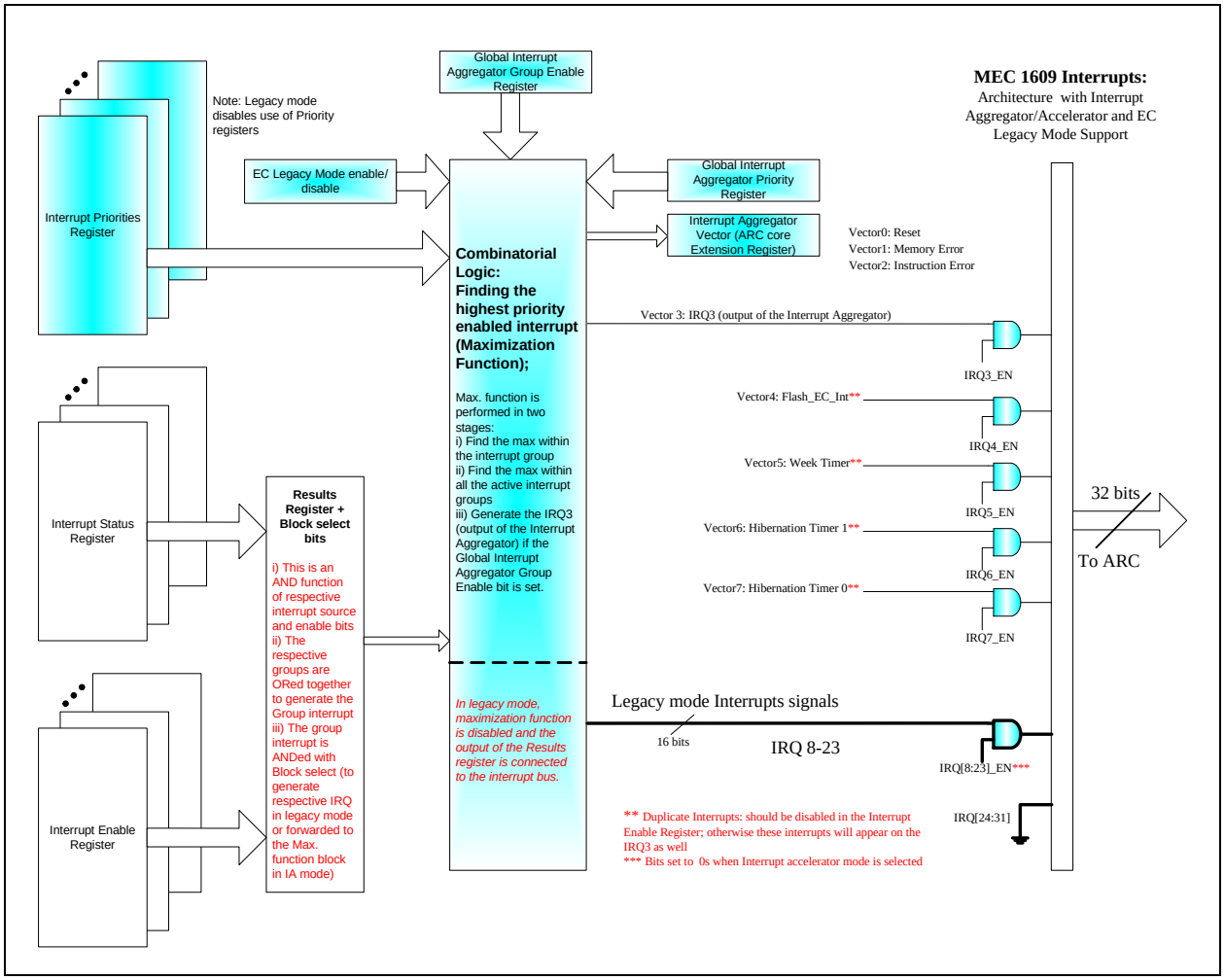
The EC Interrupt Aggregator can also operates in legacy mode to maintain compatibility with previous generation. In this mode it forwards up to 16 output interrupts to the processor's IRQ[8:23] but does not generate jump vectors.

This chapter focuses on the EC Interrupt Aggregator. Please refer to ARC International's *ARCompact™ Instruction Set Architecture Programmer's Reference*, March 2005 for more information on interrupt and exception handling by the ARC625D core.

Features of the EC Interrupt Aggregator:

- Edge-triggered inputs
- 4 priority levels
- 16 x 31 input interrupts (31 interrupts per group; 16 groups)
- Wake interrupts recognized while clock is stopped
- Programmable base address of the Vector Table
- Assist fast interrupt handling by software
  - Provides interrupt's jump vector in processor's extension register for fast access to Interrupt Vector Table
  - Support for NORM instruction to quickly locate the active highest-level interrupt

FIGURE 16-1: ARCHITECTURAL VIEW OF INTERRUPT AGGREGATOR



16.2 Interrupt Summary

Table 16-1, "EC Interrupt Structure" summarizes the ARC interrupts, priorities and vector locations.

- Link registers (ILINK1 / ILINK2) are the processor's registers that hold the value of the next PC when an interrupt occurs.
- Inside the processor, exceptions have HIGH priority and interrupts have MID or LOW priority. Within a priority level, a higher numbered interrupt has higher priority. For example, the Flash_EC_Int Interrupt with relative priority L26 has higher priority than the Week Timer's Interrupt with a relative priority L25. An exception is interrupt #7, which always has highest priority within its level.
- Byte offset: The ARC processor implements a table of jumps rather than interrupt vectors. When an interrupt occurs, the processor jumps to fixed addresses in memory, which contain a jump instruction to the interrupt handler. Byte offsets are vector offsets to the jump table.

Details on processor handling of interrupts can be found in the ARC International's *ARCompact™ Instruction Set Architecture Programmer's Reference*, March 2005.

TABLE 16-1: EC INTERRUPT STRUCTURE

Vector	Name	Link Register	Priority (Default)	Relative Priority	Byte Offset
0	Reset	-	High	H1	0x00
1	Memory Error	ILINK2	High	H2	0x08
2	Instruction Error	ILINK2	High	H3	0x10
3	Interrupt Aggregator	ILINK1	level 1 (low)	L27	0x18
4	Flash_EC_Int	ILINK1	level 1 (low)	L26	0x20
5	Week Timer	ILINK1	level 1 (low)	L25	0x28
6	Hibernation Timer 1	ILINK2	level 2 (mid)	M2	0x30
7	Hibernation Timer 0	ILINK2	level 2 (mid)	M1	0x38
8	IRQ8	ILINK1	level 1 (low)	L24	0x40
9	IRQ9	ILINK1	level 1 (low)	L23	0x48
10	IRQ10	ILINK1	level 1 (low)	L22	0x50
11	IRQ11	ILINK1	level 1 (low)	L21	0x58
12	IRQ12	ILINK1	level 1 (low)	L20	0x60
13	IRQ13	ILINK1	level 1 (low)	L19	0x68
14	IRQ14	ILINK1	level 1 (low)	L18	0x70
15	IRQ15	ILINK1	level 1 (low)	L17	0x78
16	IRQ16	ILINK1	level 1 (low)	L16	0x80
17	IRQ17	ILINK1	level 1 (low)	L15	0x88
18	IRQ18	ILINK1	level 1 (low)	L14	0x90
19	IRQ19	ILINK1	level 1 (low)	L13	0x98
20	IRQ20	ILINK1	level 1 (low)	L12	0xA0
21	IRQ21	ILINK1	level 1 (low)	L11	0xA8
22	IRQ22	ILINK1	level 1 (low)	L10	0xB0
23	IRQ23	ILINK1	level 1 (low)	L9	0xB8
24	IRQ24 (tied low)	ILINK1	level 1 (low)	L8	0xC0
25	IRQ25 (tied low)	ILINK1	level 1 (low)	L7	0xC8
26	IRQ26 (tied low)	ILINK1	level 1 (low)	L6	0xD0
27	IRQ27 (tied low)	ILINK1	level 1 (low)	L5	0xD8
28	IRQ28 (tied low)	ILINK1	level 1 (low)	L4	0xE0
29	IRQ29 (tied low)	ILINK1	level 1 (low)	L3	0xE8
30	IRQ30 (tied low)	ILINK1	level 1 (low)	L2	0xF0
31	IRQ31 (tied low)	ILINK1	level 1 (low)	L1	0xF8

16.3 Operation

16.3.1 REGISTER CONTROL OF INPUT INTERRUPTS

Associated with each interrupt are:

- a Source bit that is set to indicate when an interrupt is active
- an Interrupt Enable bit to allow interrupt generation
- a Result bit to indicate when an enabled interrupt is active
- a priority level determined by its 2 Priority Level bits

Input interrupts are organized into groups; each group comprises 31 interrupts. Associated with each group is a set of 32-bit registers (Source, Enable, Result, Priority) described above. In addition, incoming interrupts can also be controlled on group basis, i.e., all interrupts in a group can be enabled / disabled by a bit in the Group Select register. This is summarized in Table 16-2.

TABLE 16-2: INTERRUPT AGGREGATOR GROUP REGISTERS

Register Name	Width (Bits)	Purpose
Interrupt Source	31	Latches asynchronous signals from on-chip devices
Interrupt Enable	31	Enables each interrupt
Interrupt Priority	62	A 2 bit priority for each of the 31 possible interrupt sources
Interrupt Result	31	Each bit is 1 if the corresponding Interrupt Source bit is 1, the interrupt is enabled, and the priority of the interrupt is equal to or greater than the current priority. The content of this register changes continuously, i.e., combinatorially, based on outputs from the other three registers.

The 31st bit (i.e. the most significant bit) of the Result register is to control the use of the NORM instruction. should set to 0. Setting this bit to ‘0’ enables the use of the ARC NORM (normalize) instruction as a Find-First-One instruction (that is, NORM will return the bit number of the highest numbered bit that is a 1).

16.3.2 REGISTER CONTROL OF OUTPUT INTERRUPTS AND GLOBAL REGISTERS

Output interrupts to the processor, IRQ[23:03], are individually enabled. The lowest three LSBs, [2:0], are not used due the three exceptions (reset, memory error, instruction error) being directly connected to the processor.

Two global registers, IA Priority and IA Vector, are implemented as core extension registers. This means they can be used in any ARC instruction that can reference the full 6-bit core register number.

TABLE 16-3: INTERRUPT AGGREGATOR GLOBAL REGISTERS

Register Name	Width (Bits)	Purpose
IA Priority	3	The current priority level for the interrupt. A setting of 4 or higher blocks all interrupts from the Interrupt Aggregator. Interrupts at this priority level or higher will be allowed to be propagated to IRQ3. This is implemented as Core Extension register R55.
IA Group Enable	16	Enables individual Interrupt Group within IA
IA Vector	24	The address of a vector in memory. The vector is the address of an interrupt handler. for the interrupt selected by the Interrupt Aggregator. This is an ARC Core Extension Register R56.
IA IRQ Enable	20	Enable individual IRQ lines going into ARC

16.3.3 GENERATION OF INTERRUPT OUTPUTS AND JUMP VECTOR

1. The Interrupt Status register and the Interrupt Enable register respective bits are ANDed together.
2. All the bits in the same group are ORed.
3. The resultant ORed bit is ANDed with the respective Block Select Register bit to generate the IRQi in legacy mode.

4. In the accelerated mode, the results of step 3 along with the priorities set in the GIRQx Interrupt Priority Registers are fed into the Maximization Function block to generate the highest priority interrupt that is propagated to IRQ3. The address of the respective interrupt handler is subsequently loaded in the IA Vector register.
5. Prior to sending the interrupts to the ARC, IRQi_EN is ANDed with the respective IRQi.

16.3.3.1 Priority Levels

The Interrupt Aggregator adds 4 levels of interrupt priority to the 2 levels the ARC provides. Each of the potential chip interrupts (16 interrupt source registers times up to 31 sources per register) can independently be assigned one of 4 priority levels. The Interrupt Priority Register in each group has a 2-bit priority field for each of the 31 possible interrupt sources. The 3-bit IA Priority Register sets a current priority level for all groups.

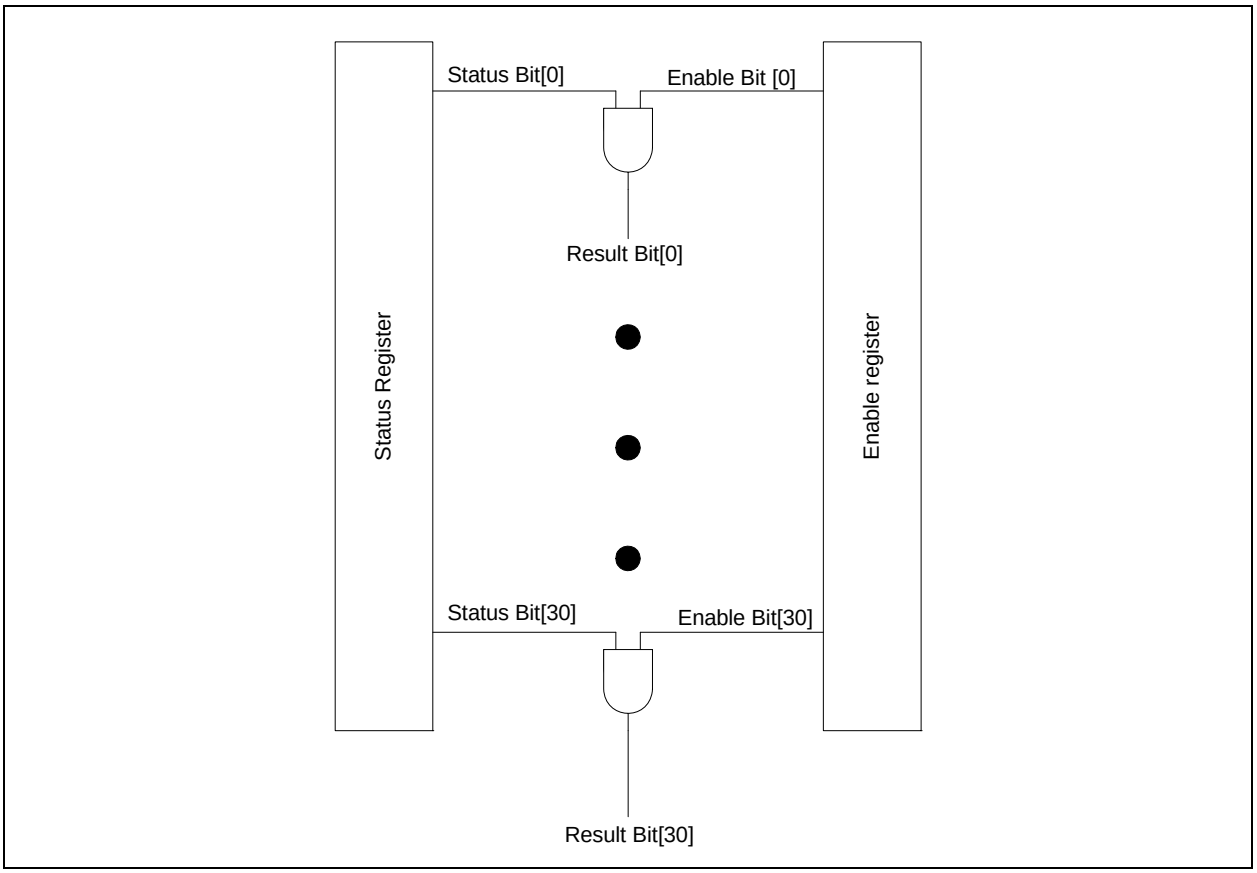
An individual interrupt *i* is enabled if bit *i* in the Interrupt Source register is 1 (asserted), bit *i* in the Interrupt Enable register is 1 (enabled), and the 2 bits for interrupt *i* in the Interrupt Priority Register represent a number that is greater than or equal to the IA Priority Register as well as greater or equal to the priority level of any other interrupt that is currently enabled.

The MEC1609 Interrupt Aggregator selects the interrupt with the highest priority among all active interrupts. Interrupts at priorities below the current are blocked. Within a group the interrupt with the higher bit number has priority over interrupts with lower bit number (assuming that these bits are at the same priority level).

16.3.3.2 Interrupt 'Result'

As shown in Figure 16-1 the Source and Enable bits are latched, but the Result bits are not. The latter change combinationally with inputs Source and Enable inputs.

FIGURE 16-2: EXAMPLE OF THE RESULTS REGISTER



16.3.3.3 Group Interrupt Request

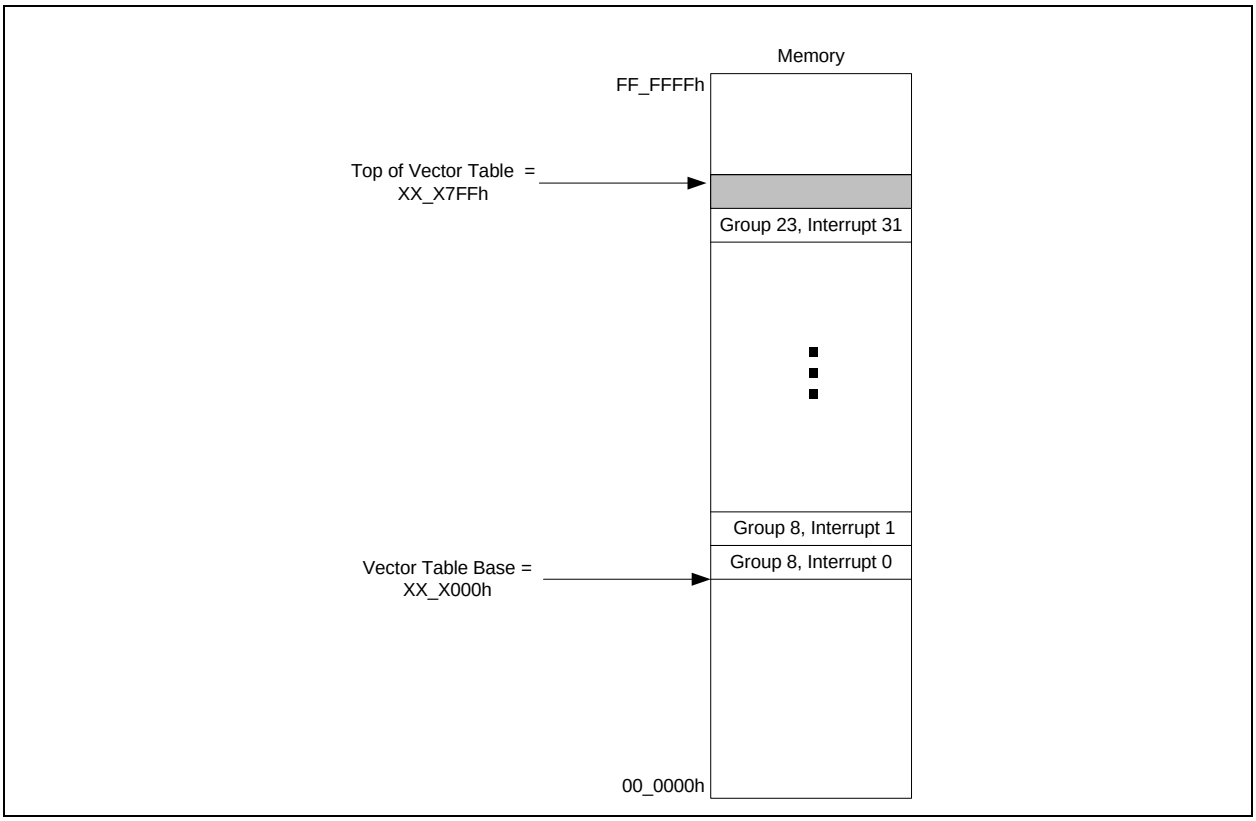
There are 16 Group Interrupt Request signals, one for each bank in the Interrupt Aggregator. An interrupt is propagated within the interrupt chain (that results in the IRQ3) if the priority assigned to the source bit in the Group Priority Register is greater than or equal to the contents of the IA Priority Register (Current Priority bits). For example, assigning a priority of 3 to a source bit means maximum priority for that source, which will always be enabled if the corresponding enable bit is set and the IA Priority Register is 3 or less. Setting the IA Priority Register to 4 or greater disables all interrupts in the Interrupt Aggregator. Software must maintain the IA Priority Register, and stack the value in memory if nested interrupts are required.

16.3.3.4 Interrupt Vector Generation

The Interrupt Aggregator continually selects from among all of its active inputs to generate one IRQ that is connected to the ARC interrupt controller on ARC input IRQ3. At the same time, the Aggregator generates an index into an Interrupt Vector table which addresses a pointer to the handler for the interrupt that is to be serviced.

The Interrupt Aggregator Interrupt Vector Table is a table of 4-byte addresses that is 2KB in length. will be used(16 Groups times 31 interrupts per group times 4 bytes per address). FIGURE 16-3: Interrupt Aggregator Vector Table on page 256 illustrates the Interrupt Vector Table:

FIGURE 16-3: INTERRUPT AGGREGATOR VECTOR TABLE

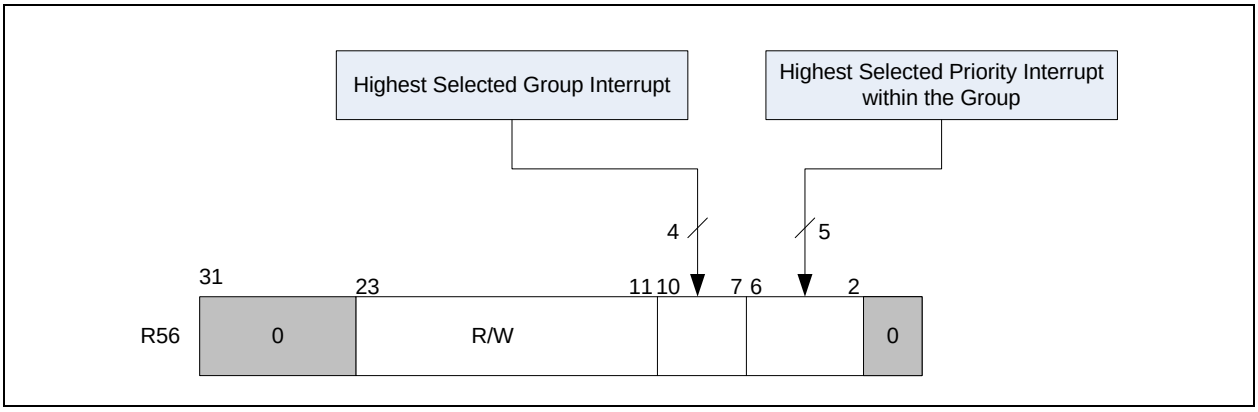


The table must begin on a 2KB address boundary. Since there are only 31 possible interrupt sources per Group Interrupt Request, the 32nd vector entry is always null. The table will typically reside in the program Flash memory.

A new extension register is added to the ARC register set to support interrupt vectoring. Register R56 is a pointer into the Interrupt Vector table and may be used in any ARC instruction that can reference one of the general purpose registers. Bits 23 through 11 of R56 are readable and writable; setting this range establishes the base address of the table. Bits 10 through 0 are read-only, which is the reason the vector table must start on a 2KB address boundary. Since ARC addresses are 16 bits, bits 31 through 24 of R56 are reserved and always read as 0.

The FIGURE 16-4: Interrupt Vector Generation on page 257 illustrates the mechanism by which bits 10 through 2 of R56 are set:

FIGURE 16-4: INTERRUPT VECTOR GENERATION



A Highest Group Interrupt block selects the highest numbered GIRQ currently active and enabled. It should be noted that even though the group numbering ranges from GIRQ8-GIRQ22, there are four bits allocated to represent these numbers. For example, if GIRQ8 is selected then value 0x0 will be assigned to bits [10:7] of the R56. Likewise, if GIRQ22 is selected, value 0xE will be assigned to bits [10:7] of R56. Table 16-4, "GIRQi Mapping to Bits [10:7] of R56" shows the mapping of the GIRQi to the bits [10:7] bits of R56.

TABLE 16-4: GIRQI MAPPING TO BITS [10:7] OF R56

$GIRQ_i$	Mapped Bit Values in the Group Select Field
GIRQ8	0000
GIRQ9	0001
GIRQ10	0010
GIRQ11	0011
GIRQ12	0100
GIRQ13	0101
GIRQ14	0110
GIRQ15	0111
GIRQ16	1000
GIRQ17	1001
GIRQ18	1010
GIRQ19	1011
GIRQ20	1100
GIRQ21	1101
GIRQ22	1110

The “Highest Priority Interrupt within the Group” block selects the highest numbered interrupt within the group that is active and enabled. The five bit result becomes bits 6 through 2 of R56. The low two bits of R56 are always 0, since the register always points to a word-aligned address.

A two instruction sequence is sufficient to vector to one of the Interrupt Aggregator interrupts. The following two-instruction sequence fits into the IRQ3 8-byte vector slot used by the ARC:

```
LDRx,[R56]              ;fetch the address for the active interrupt
J[Rx]                   ;Jump to handler
```

The register Rx should be reserved for exclusive use by the interrupt handler, using either Register R53 or R54, or using compiler option -Hreg_reserve=x, so that the register is never live an interrupt is serviced.

An alternate ARC handler would be:

```
JCommon_Int_Handler     ;Jump to handler
```

where `Common_Int_Handler` is a routine that handles interrupt state, performs any interrupt stack maintenance, and sets up whatever is required for the actual interrupt handler. The jump instruction is 8 bytes long and can reach any address in the ARC address space. When the common code is finished, the same two instruction sequence listed above can be used to jump to the correct interrupt handler.

The minimal ARC vector would consist of one 8-byte jump instruction. The time penalty for adding the load from R56 is two cycles, one for the standard ARC load delay and an additional cycle the ARC requires when fetching data from instruction space.

16.3.4 NON MASKABLE INTERRUPTS

The ARC does not have a non maskable interrupt input. It is straightforward to assign one of the Hibernation or Week timers, which can generate ARC interrupts on ARC IRQ5 through IRQ7, to ARC priority level 2, and all other interrupts to priority level 1. Since no other interrupt handler should ever disable interrupts on Priority 2, the timer interrupt will always be enabled.

It may not be possible to use a non maskable interrupt when the flash memory is being programmed by the Host. The Flash interrupt is asserted when a Host has completed Flash programming and the flash array is again available to the ARC. While the flash is being programmed, the ARC has no program space (which is normally in the flash), and so must sleep. Since it has no vector table, it cannot respond to interrupts while the flash is busy. All interrupts except Flash_EC_Int must be disabled; the Flash interrupt can be enabled since it is only asserted when the flash is again available to the ARC. One simple way to ensure this is to assign the Flash_EC_Int interrupt Priority Level 2, assign all other ARC IRQs to Priority Level 1, and disable Priority Level 1 before putting the ARC to sleep. An alternative would be to remap the ARC Interrupt Vector Base Register to the SRAM and define an interrupt vector table, along with its handlers, in the SRAM.

16.3.5 WAKE CAPABLE INTERRUPTS

The EC Interrupt Aggregator routes logic from WAKE Event Sources to the WAKE input of the Power, Clocks and Resets-Power Management Interface to wake the system. This logic requires no clocks.

The interrupt sources AND’ed with the corresponding Enable bit will be OR’ed to produce a wake event.

The wake up sources are identified in a “WAKE” column of the Bit definitions table for each IRQ’s Source Register.

16.3.6 NON-WAKE CAPABLE INTERRUPTS

These interrupts require a running clock in their source block to be recognized and presented to the interrupt aggregator. Please consult the WAKE column of the Bit definitions table for each IRQ’s Source Register.

16.3.7 INTERRUPTS DIRECTLY CONNECTED TO ARC PROCESSOR

Interrupts from the two Hibernation Timers, the Week Timer and the Embedded Flash interface are routed to the EC Interrupt lines IRQ7 through IRQ4, respectively.

PROGRAMMER’S NOTE:	In non-legacy mode the Hibernation Timers, the Week Timer and the Embedded Flash interface are directly connected to the interrupt line and should be disabled in the Interrupt Enable Register. Otherwise these interrupts will be duplicated on the IRQ3 as well as their respective interrupt lines.

16.3.8 DISABLING INTERRUPTS

Because of pipeline latency, delay through the Load/Store queue and traffic on the AHB bus, writes to SPB registers can potentially take many processor cycles to complete. Because of this latency, the IA Group Select Register register and the IRQ Enable Register should not be used for disabling interrupts for software operations like critical sections. Several interrupts could potentially fire between the STORE instruction to the interrupt registers and the instruction after the STORE.

The ARC FLAG instruction is used to modify the E1 and E2 interrupt enable bits in the STATUS32 register. If the FLAG instruction is used, software can insure that no unexpected interrupts will be processed in the middle of a critical section. The following example illustrates how the FLAG instruction might be used to implement a critical section:

```
FLAG; 0                 ; disable all interrupts
NOP                     ; Pipeline Flush
NOP                     ; Pipeline Flush
NOP                     ; Pipeline Flush
;
; <critical section code here>
;
FLAG; 6;                ; enable all interrupts
```

16.4 Power, Clocks and Reset

16.4.1 POWER DOMAIN

This block is powered by VTR for wake up capability.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

16.4.2 CLOCKS

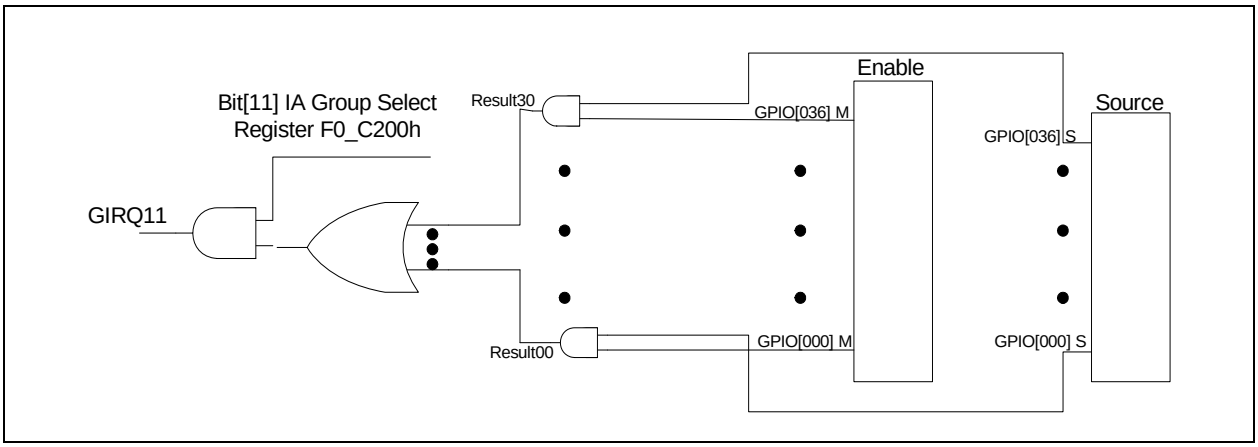
Use MCLK.

16.4.3 RESET

This block is reset by nSYS_RST. Following a reset, Interrupt Source, Enable and Result registers default to ‘0’ and all interrupts are enabled.

16.4.4 INTERRUPT ROUTING

FIGURE 16-5: GPIO INTERRUPT STRUCTURE EXAMPLE



16.5 Registers Overview

16.5.1 ADDRESSING

The EC Interrupt Aggregator has its own Logical Device Number, and Base Address as indicated in Table 16-5.

TABLE 16-5: EC Interrupt Aggregator BASE ADDRESS TABLE

Block	LDN	AHB Base Address
EC Interrupt Aggregator	30h	F0_C000h

16.5.2 REGISTERS SUMMARY

Table 16-6 is a register summary for the EC Interrupt Aggregator block. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 16-6: EC Interrupts REGISTER SUMMARY

	EC Interface			Notes	
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Bit Definitions**	**Logical Devices**
GIRQ8 Source Register	00h	3-0	R/WC	Table 16-9	GPIO140-GPIO176
GIRQ8 Enable Register	04h	3-0	R/W		
GIRQ8 Result Register	08h	3-0	R		
GRIQ8a Priority Register	0Ch	3-0	R/W	Table 16-7	
GRIQ8b Priority Register	10h	3-0	R/W	Table 16-8	
GIRQ9 Source Register	14h	3-0	R/WC	Table 16-13	GPIO100-GPIO136
GIRQ9 Enable Register	18h	3-0	R/W		
GIRQ9 Result Register	1Ch	3-0	R		
GRIQ9a Priority Register	20h	3-0	R/W	Table 16-7	
GRIQ9b Priority Register	24h	3-0	R/W	Table 16-8	
GIRQ10 Source Register	28h	3-0	R/WC	Table 16-17	GPIO040-GPIO076
GIRQ10 Enable Register	2Ch	3-0	R/W		
GIRQ10 Result Register	30h	3-0	R		
GRIQ10a Priority Register	34h	3-0	R/W	Table 16-7	
GRIQ10b Priority Register	38h	3-0	R/W	Table 16-8	
GIRQ11 Source Register	3Ch	3-0	R/WC	Table 16-22	GPIO000-GPIO036
GIRQ11 Enable Register	40h	3-0	R/W		
GIRQ11 Result Register	44h	3-0	R		
GRIQ11a Priority Register	48h	3-0	R/W	Table 16-7	
GRIQ11b Priority Register	4Ch	3-0	R/W	Table 16-8	
GIRQ12 Source Register	50h	3-0	R/WC	Table 16-26	SMBus
GIRQ12 Enable Register	54h	3-0	R/W		
GIRQ12 Result Register	58h	3-0	R		
GRIQ12a Priority Register	5Ch	3-0	R/W	Table 16-7	
GRIQ12b Priority Register	60h	3-0	R/W	Table 16-8	
GIRQ13 Source Register	64h	3-0	R/WC	Table 16-30	ACPI PM1
GIRQ13 Enable Register	68h	3-0	R/W		
GIRQ13 Result Register	6Ch	3-0	R		

TABLE 16-6: EC Interrupts REGISTER SUMMARY (CONTINUED)

	EC Interface			Notes	
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Bit Definitions**	**Logical Devices**
GRIQ13a Priority Register	70h	3-0	R/W	Table 16-7	
GRIQ13b Priority Register	74h	3-0	R/W	Table 16-8	
GIRQ14 Source Register	78h	3-0	R/WC	Table 16-34	LPC Interface, EC GP-SPI, Embedded Flash
GIRQ14 Enable Register	7Ch	3-0	R/W		
GIRQ14 Result Register	80h	3-0	R		
GRIQ14a Priority Register	84h	3-0	R/W	Table 16-7	
GRIQ14b Priority Register	88h	3-0	R/W	Table 16-8	
GIRQ15 Source Register	8Ch	3-0	R/WC	Table 16-38	Mailbox Interface, UART
GIRQ15 Enable Register	90h	3-0	R/W		
GIRQ15 Result Register	94h	3-0	R		
GRIQ15a Priority Register	98h	3-0	R/W	Table 16-7	
GRIQ15b Priority Register	9Ch	3-0	R/W	Table 16-8	
GIRQ16 Source Register	A0h	3-0	R/WC	Table 16-42	RC ID, ADC, PECI
GIRQ16 Enable Register	A4h	3-0	R/W		
GIRQ16 Result Register	A8h	3-0	R		
GRIQ16a Priority Register	ACh	3-0	R/W	Table 16-7	
GRIQ16b Priority Register	B0h	3-0	R/W	Table 16-8	
GIRQ17 Source Register	B4h	3-0	R/WC	Table 16-46	TACH
GIRQ17 Enable Register	B8h	3-0	R/W		
GIRQ17 Result Register	BCh	3-0	R		
GRIQ17a Priority Register	C0h	3-0	R/W	Table 16-7	
GRIQ17b Priority Register	C4h	3-0	R/W	Table 16-8	
GIRQ18 Source Register	C8h	3-0	R/WC	Table 16-50	BC Bus Master
GIRQ18 Enable Register	CCh	3-0	R/W		
GIRQ18 Result Register	D0h	3-0	R		
GRIQ18a Priority Register	D4h	3-0	R/W	Table 16-7	
GRIQ18b Priority Register	D8h	3-0	R/W	Table 16-8	
GIRQ19 Source Register	DCh	3-0	R/WC	Table 16-54	Keyboard Controller (8042), PS/2
GIRQ19 Enable Register	E0h	3-0	R/W		
GIRQ19 Result Register	E4h	3-0	R		
GRIQ19a Priority Register	E8h	3-0	R/W	Table 16-7	
GRIQ19b Priority Register	ECh	3-0	R/W	Table 16-8	
GIRQ20 Source Register	F0h	3-0	R/WC	Table 16-58	ACPI EC MSG
GIRQ20 Enable Register	F4h	3-0	R/W		
GIRQ20 Result Register	F8h	3-0	R		
GRIQ20a Priority Register	FCh	3-0	R/W	Table 16-7	
GRIQ20b Priority Register	100h	3-0	R/W	Table 16-8	

TABLE 16-6: EC Interrupts REGISTER SUMMARY (CONTINUED)

	EC Interface			Notes	
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Bit Definitions**	**Logical Devices**
GIRQ21 Source Register	104h	3-0	R/WC	Table 16-62	VLPC Interface
GIRQ21 Enable Register	108h	3-0	R/W		
GIRQ21 Result Register	10Ch	3-0	R		
GRIQ21a Priority Register	110h	3-0	R/W	Table 16-7	
GRIQ21b Priority Register	114h	3-0	R/W	Table 16-8	
GIRQ22 Source Register	118h	3-0	R/WC	Table 16-66	GPIO[217:200]
GIRQ22 Enable Register	11Ch	3-0	R/W		
GIRQ22 Result Register	120h	3-0	R		
GRIQ22a Priority Register	124h	3-0	R/W	Table 16-7	
GRIQ22b Priority Register	128h	3-0	R/W	Table 16-8	
GIRQ23 Source Register	12Ch	3-0	R/WC	Table 16-70	Week Alarm Timer, 16-bit Timer Hibernation Timer Input Capture and Compare Timer
GIRQ23 Enable Register	130h	3-0	R/W		
GIRQ23 Result Register	134h	3-0	R		
GRIQ23a Priority Register	138h	3-0	R/W	Table 16-7	
GRIQ23b Priority Register	13Ch	3-0	R/W	Table 16-8	
IA Group Select Register	200h	3-0	R/W	Table 16-73	
IRQ Enable Register	204h	3-0	R/W	Table 16-74	

16.6 Register Descriptions

Input interrupts are grouped into groups of up to 31 interrupts each. Associated with each group is a set of Source, Enable, Priority, and Result registers. Registers are 32-bit. There are two Priority registers for each group since an interrupt's priority level is encoded by 2 bits.

The aforementioned four register 'types' are first described in generic terms. Subsequent sections describe register bits in terms of specific interrupts they are associated with.

16.6.1 GIRQX SOURCE REGISTERS

There are 16 Group Source Enable registers, one per Interrupt Group.

Int_Source[30:0]:

A bit in this field is set when the corresponding interrupt input is active. The Interrupt Aggregator recognizes level-triggered, active-high inputs. Other input types are to be captured and relayed to the Aggregator. For example, the GPIO interface can register external edge-triggered interrupts and forward them to the Aggregator as active-high, level interrupts.

16.6.2 GIRQX ENABLE REGISTERS

There are 16 Group Interrupt Enable registers, one per Interrupt Group.

Int_Enable[30:0]:

Each bit in this field enables an interrupt from the like-numbered bit in the associated Interrupt Source register. Interrupt *i* is disabled if Int_Enable[*i*] is 0 and enabled if Int_Enable[*i*] is 1 and the Int_Priority[*i*] is greater than or equal to the current priority level. See Table 16-6, "EC Interrupts Register Summary" for EC Offset addresses for the 16 GIRQx Enable registers.

16.6.3 GIRQX A PRIORITY REGISTERS

There are 16 Group Interrupt A Priority registers, one per Interrupt Group. Group Interrupt A Priority registers, combined with the Group Interrupt B Priority registers, determine the 2-bit priority level for all interrupts in an Interrupt Group. The format of all A Priority registers is the same, described in Table 16-7, "GIRQx A Priority Register": *It should be noted that at times not all corresponding source register bits are used, hence the respective Int_priority bits should not be populated.*

TABLE 16-7: GIRQX A PRIORITY REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	xxh				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority15		Int_Priority14		Int_Priority13		Int_Priority12	
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority11		Int_Priority10		Int_Priority9		Int_Priority8	
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority7		Int_Priority6		Int_Priority5		Int_Priority4	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority3		Int_Priority2		Int_Priority1		Int_Priority0	

INT_PRIORITY[15:0]

Each of the 16 2-bit fields in this register sets the priority level for interrupts assigned to Interrupt Source register bit 15 through bit 0, in the same Interrupt Group. See Table 16-6, "EC Interrupts Register Summary" for EC Offset addresses for the sixteen GIRQx A Priority registers.

16.6.4 GIRQX B PRIORITY REGISTERS

There are 16 Group Interrupt B Priority registers, one per Interrupt Group. Group Interrupt B Priority registers, combined with the Group Interrupt A Priority registers, determine the 2-bit priority level for all interrupts in an Interrupt Group. The format of all B Priority registers is the same, described in Table 16-8, "GIRQx B Priority Register":*It should be noted that at times not all corresponding source register bits are used, hence the respective Int_priority bits should not be populated.*

TABLE 16-8: GIRQX B PRIORITY REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	xxh			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		Int_Priority30		Int_Priority29		Int_Priority28	
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority27		Int_Priority26		Int_Priority25		Int_Priority24	
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority23		Int_Priority22		Int_Priority21		Int_Priority20	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority19		Int_Priority18		Int_Priority17		Int_Priority16	

INT_PRIORITY[30:16]

Each of the 16 2-bit fields in this register sets the priority level for interrupts assigned to Interrupt Source register bit 30 through bit 16, in the same Interrupt Group. See Table 16-6, "EC Interrupts Register Summary" for EC Offset addresses for the sixteen GIRQx B Priority registers.

16.6.5 GIRQX RESULT REGISTERS

There are 16 Group Interrupt Result registers, one per Interrupt Group.

INT_RESULT[30:0]

Each bit in this field is 1 if an interrupt from the like-numbered bit in the associated Interrupt Source register is active. Interrupt *i* in each Interrupt Group is active if and only if Int_Source[*i*] is 1, Int_Enable[*i*] is 1 and Int_Priority[*i*] is greater than or equal to IA Priority register as well as any enabled interrupt in all Interrupt Groups. The GIRQx Result Register is not latched but is a function of Int_Source, Int_Enable and Int_Priority. See Figure 16-2 for an explanation of how Result is generated.

16.6.6 GIRQ8 SOURCE REGISTER

TABLE 16-9: GIRQ8 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	00h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-10 & Note 16-1 on page 265						
BIT NAME	Reserved	GPIO[140:176]						

TABLE 16-10: BIT DEFINITIONS GIRQ8 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[147:140]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect.
[15:8]	R/WC	GPIO[157:150]	Y	
[20:16]	R/WC	GPIO[164:160]	Y	
[23:21]	R	GPIO[167:165]	N	Reserved
[30:24]	R	GPIO[176:170]	N	Reserved
31	R	Reserved	N	This bit is always reserved

Note 16-1 Reserved Bits shown in the Source register are also reserved read only in the Corresponding Enable & Result register.

16.6.7 GIRQ8 ENABLE REGISTER

TABLE 16-11: GIRQ8 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	04h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 16-10 & Note 16-1 on page 265						
BIT NAME	Reserved	GPIO[140:176]						

16.6.8 GIRQ8 RESULT REGISTER

TABLE 16-12: GIRQ8 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	08h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[140:176]						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.9 GIRQ9 SOURCE REGISTER

TABLE 16-13: GIRQ9 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	14h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-14 & Note 16-1 on page 265						
BIT NAME	Reserved	GPIO[100:136]						

TABLE 16-14: GBIT DEFINITIONS IRQ9 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[107:100]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect.
[15:8]	R/WC	GPIO[117:110]	Y	
[23:16]	R/WC	GPIO[127:120]	Y	
[26:24]	R/WC	GPIO[132:130]	Y	
[30:27]	R	GPIO[136:133]	N	Reserved
31	R	Reserved	N	This bit is always reserved.

16.6.10 GIRQ9 ENABLE REGISTER

TABLE 16-15: GIRQ9 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	18h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 16-14 & Note 16-1 on page 265.						
BIT NAME	Reserved	GPIO[100:136]						

16.6.11 GIRQ9 RESULT REGISTER

TABLE 16-16: GIRQ9 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	1Ch					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[100:136]						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.12 GIRQ10 SOURCE REGISTER

TABLE 16-17: G IRQ10 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	28h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-18 & Note 16-1 on page 265.						
BIT NAME	Reserved	GPIO[040:076]						

TABLE 16-18: BIT DEFINITIONS GIRQ10 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[047:040]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect.
[15:8]	R/WC	GPIO[057:050]	Y	
[23:16]	R/WC	GPIO[067:060]	Y	
[30:24]	R	GPIO[076:070]	N	Reserved
31	R	Reserved	N	This bit is always reserved

16.6.13 GIRQ10 ENABLE REGISTER

TABLE 16-19: G IRQ10 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	2Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 16-18 & Note 16-1 on page 265.						
BIT NAME	Reserved	GPIO[040:076]						

16.6.14 GIRQ10 RESULT REGISTER

TABLE 16-20: GIRQ10 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	30h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[040:076]						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.15 GIRQ11 SOURCE REGISTER

TABLE 16-21: GIRQ11 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	3Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-22 & Note 16-1 on page 265.						
BIT NAME	Reserved	GPIO[000:036]						

TABLE 16-22: BIT DEFINITIONS GIRQ11 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[007:000]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value ‘1’ to this bit. Write to this bit with a value of ‘0’ have no effect.
[15:8]	R/WC	GPIO[017:010]	Y	
[23:16]	R/WC	GPIO[027:020]	Y	
[26:24]	R/WC	GPIO[032:030]	Y	
[30:27]	R	GPIO[036:033]	N	Reserved
31	R	Reserved	N	This bit is always reserved

16.6.16 GIRQ11 ENABLE REGISTER

TABLE 16-23: GIRQ11 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	40h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 16-22 & Note 16-1 on page 265.						
BIT NAME	Reserved	GPIO[000:036]						

16.6.17 GIRQ11 RESULT REGISTER

TABLE 16-24: GIRQ11 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	44h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[000:036]						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.18 GIRQ12 SOURCE REGISTER

TABLE 16-25: GIRQ12 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	50h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-26 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-26, “Bit Definitions GIRQ12 Source Register,” on page 270.						

TABLE 16-26: BIT DEFINITIONS GIRQ12 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	SMB0	N	I2C/SMBus controller 0 interrupt. This interrupt is signaled when the I2C/SMBus controller 0 asserts its interrupt request
1	SMB1	N	I2C/SMBus controller 1 interrupt. This interrupt is signaled when the I2C/SMBus controller 1 asserts its interrupt request
2	SMB2	N	I2C/SMBus controller 2 interrupt. This interrupt is signaled when the I2C/SMBus controller 2 asserts its interrupt request
3	SMB00 WK	Y	I2C/SMBus Port 00 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB00.
4	SMB01 WK	Y	I2C/SMBus Port 01 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB01.
5	SMB02 WK	Y	I2C/SMBus Port 02 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB02.
6	SMB03 WK	Y	I2C/SMBus Port 03 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB03.
7	SMB04 WK	Y	I2C/SMBus Port 04 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB04.

TABLE 16-26: BIT DEFINITIONS GIRQ12 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
8	SMB05 WK	Y	I2C/SMBus Port 05 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB05.
9	SMB06 WK	Y	I2C/SMBus Port 06 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB06.
10	SMB07 WK	Y	I2C/SMBus Port 07 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB07.
11	SMB10 WK	Y	I2C/SMBus Port 10 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB10.
12	SMB11 WK	Y	I2C/SMBus Port 11 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB11.
13	SMB12 WK	Y	I2C/SMBus Port 12 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB12.
14	SMB20 WK	Y	I2C/SMBus Port 20 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB20.
15	SMB21 WK	Y	I2C/SMBus Port 21 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB21.
16	SMB22 WK	Y	I2C/SMBus Port 22 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB22.
17	SB_TSI	Y	SB-TSI (Port 23) wake interrupt.
30-18	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-2 Source bits have corresponding Enable and Result bits.

16.6.19 GIRQ12 ENABLE REGISTER

TABLE 16-27: GIRQ12 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	54h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-26 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-26, “Bit Definitions GIRQ12 Source Register,” on page 270.						

16.6.20 GIRQ12 RESULT REGISTER

TABLE 16-28: GIRQ12 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	58h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-26, “Bit Definitions GIRQ12 Source Register,” on page 270.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.21 GIRQ13 SOURCE REGISTER

TABLE 16-29: GIRQ13 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	64h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-30 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-30, “Bit Definitions GIRQ13 Source Register,” on page 272.						

TABLE 16-30: BIT DEFINITIONS GIRQ13 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	PM1_CTL2	N	PM1_CTL2 written by Host
1	PM1_EN2	N	PM1_EN2 written by Host
2	PM1_STS2	N	PM1_STS2 written by Host
15:3	Reserved	N	Reserved
16	DMA_7	N	DMA Channel 7
17	DMA_6	N	DMA Channel 6
18	DMA_5	N	DMA Channel 5
19	DMA_4	N	DMA Channel 4
20	DMA_3	N	DMA Channel 3
21	DMA_2	N	DMA Channel 2
22	DMA_1	N	DMA Channel 1

TABLE 16-30: BIT DEFINITIONS GIRQ13 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
23	DMA_0	N	DMA Channel 0
30-24	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-3 Source bits have corresponding Enable and Result bits.

16.6.22 GIRQ13 ENABLE REGISTER

TABLE 16-31: GIRQ13 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	68h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W except for reserved bits See Table 16-30 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-30, “Bit Definitions GIRQ13 Source Register,” on page 272.						

16.6.23 GIRQ13 RESULT REGISTER

TABLE 16-32: GIRQ13 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	6Ch					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-30, “Bit Definitions GIRQ13 Source Register,” on page 272.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.24 GIRQ14 SOURCE REGISTER

TABLE 16-33: GIRQ14 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	78h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-34 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-34, “Bit Definition GIRQ14 source Register,” on page 274.						

TABLE 16-34: BIT DEFINITION GIRQ14 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	LPCPD#	Y	LPC Power Down pin state
1	LRESET#	Y	LRESET Interrupt. This interrupt is signaled when LRESET is asserted.
2	LPC_AHB_ERR	N	either a LPC BAR conflict or an AHB bus error occurred as a result of an LPC access.
3	SPI_TXBE_GP	N	Tx buffer empty from the GP_SPI block on EC AHB
4	SPI_RXBF_GP	N	Rx buffer bfull from the GP_SPI block on EC AHB
5	FLASH_BUSY_ERR	N	Embedded Flash Busy Error
6	FLASH_CMD_ERR	N	Embedded Flash Command Error
7	FLASH_PROTECT_ERR	N	Embedded Flash Protect Error
8	FLASH_EC_INT	N	Host-to-EC Interrupt that transfers control of the Flash to the EC
9	VCC_PWRGD_INT	Y	VCC_PWRGD (from GPIO 57)
15:10	Reserved	-	Reserved
16	GP_SPI_TXBE	N	Tx buffer empty from the GP_SPI block on LPC AHB
17	GP_SPI_RXBF	N	Rx buffer bfull from the GP_SPI block on LPC AHB
18	ASIF_INT	N	Interrupt from the ASIF block’s EC logical interface
30-19	Reserved	-	Reserved
31	Reserved	-	This bit is always reserved

Note 16-4 Source bits have corresponding Enable and Result bits.

16.6.25 GIRQ14 ENABLE REGISTER

TABLE 16-35: GIRQ14 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	7Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-34 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-34, “Bit Definition GIRQ14 source Register,” on page 274.						

16.6.26 GIRQ14 RESULT REGISTER

TABLE 16-36: GIRQ14 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	80h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-34, “Bit Definition GIRQ14 source Register,” on page 274.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.27 GIRQ15 SOURCE REGISTER

TABLE 16-37: GIRQ15 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	8Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-38 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-38, “Bit Definition IRQ15 Source Register,” on page 276.						

TABLE 16-38: BIT DEFINITION IRQ15 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	UART_RX	N	UART Interrupt
1	MBX	N	Mailbox Register Interface EC Interrupt
2	EM_MBX	N	Embedded Memory Interface Host-to-EC Mailbox Interrupt
30-3	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-5 Source bits have corresponding Enable and Result bits.

16.6.28 GIRQ15 ENABLE REGISTER

TABLE 16-39: GIRQ15 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	90h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-38 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-38, “Bit Definition IRQ15 Source Register,” on page 276.						

16.6.29 GIRQ15 RESULT REGISTER

TABLE 16-40: GIRQ15 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	94h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-38, “Bit Definition IRQ15 Source Register,” on page 276.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.30 GIRQ16 SOURCE REGISTER

TABLE 16-41: GIRQ16 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	A0h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-42 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-42, “Bit Definition IRQ16 Source Register,” on page 277.						

TABLE 16-42: BIT DEFINITION IRQ16 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	RCID	N	0-to-1 transition of RC_ID done flag
1	ADC_ONESTAT	N	ADC’s one-shot conversion completion interrupt
2	ADC_RTPSTAT	N	ADC’s repeated conversion interrupt
3	PECI_INT	N	PECI interrupt
30-4	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-6 Source bits have corresponding Enable and Result bits.

16.6.31 GIRQ16 ENABLE REGISTER

TABLE 16-43: GIRQ16 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	A4h					32-bit	**EC SIZE**	
POWER	VTR					0000_00000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-42 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-42, “Bit Definition IRQ16 Source Register,” on page 277.						

16.6.32 GIRQ19 RESULT REGISTER

TABLE 16-44: GIRQ16 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	A8h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-42, “Bit Definition IRQ16 Source Register,” on page 277.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.33 GIRQ17 SOURCE REGISTER

TABLE 16-45: GIRQ17 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	B4h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-46 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-46, “Bit Definition GIRQ17 Source Register,” on page 278.						

TABLE 16-46: BIT DEFINITION GIRQ17 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	TACH0	N	Fan TACH 0 Interrupt.
1	TACH1	N	Fan TACH 1 Interrupt.
2	TACH2	N	Fan TACH 2 Interrupt.
3	TACH3	N	Fan TACH 3 Interrupt.
30-4	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-7 Source bits have corresponding Enable and Result bits.

16.6.34 GIRQ17 ENABLE REGISTER

TABLE 16-47: GIRQ17 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	B8h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-46 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-46, “Bit Definition GIRQ17 Source Register,” on page 278.						

16.6.35 GIRQ17 RESULT REGISTER

TABLE 16-48: GIRQ17 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	98h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-46, “Bit Definition GIRQ17 Source Register,” on page 278.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.36 GIRQ18 SOURCE REGISTER

TABLE 16-49: GIRQ18 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	C8h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-50 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-50, “Bit Definition IRQ18 Source Register,” on page 280.						

TABLE 16-50: BIT DEFINITION IRQ18 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	BCM_BUSY_CLR[A]	N	BC-LINK Busy Clear Flag Interrupt
1	BCM_ERR[A]	N	BC_LINK Error Flag Interrupt
2	BCM_INT#[A]	Y	Interrupt from the BC_LINK Companion
3	BCM_BUSY_CLR[B]	N	BC-LINK Busy Clear Flag Interrupt
4	BCM_ERR[B]	N	BC_LINK Error Flag Interrupt
5	BCM_INT#[B]	Y	Interrupt from the BC_LINK Companion
6	BCM_BUSY_CLR[C]	N	BC-LINK Busy Clear Flag Interrupt
7	BCM_ERR[C]	N	BC_LINK Error Flag Interrupt
8	BCM_INT#[C]	Y	Interrupt from the BC_LINK Companion
9	BCM_BUSY_CLR[D]	N	BC-LINK Busy Clear Flag Interrupt
10	BCM_ERR[D]	N	BC_LINK Error Flag Interrupt
11	BCM_INT#[D]	Y	Interrupt from the BC_LINK Companion
15-12	Reserved	N	Reserved
16	KEYSCAN	Y	Interrupt from keyscan block
30-17	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-8 Source bits have corresponding Enable and Result bits.

16.6.37 GIRQ18 ENABLE REGISTER

TABLE 16-51: GIRQ18 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	CCh					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-50 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-50, “Bit Definition IRQ18 Source Register,” on page 280.						

16.6.38 GIRQ18 RESULT REGISTER

TABLE 16-52: GIRQ18 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	D0h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-50, “Bit Definition IRQ18 Source Register,” on page 280.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.39 GIRQ19 SOURCE REGISTER

TABLE 16-53: GIRQ19 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	DCh					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-54 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-54, “Bit Definition GIRQ19 Source Register,” on page 281.						

TABLE 16-54: BIT DEFINITION GIRQ19 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	KBD_OBF	N	Keyboard Controller OBF Interrupt. This interrupt is signaled when the OBF bit in the KBD status register has been clear.
1	KBD_IBF	N	Keyboard Controller IBF Interrupt. This interrupt is signaled when the host writes to the KBD Command or Data port.
12-2	Reserved	N	Reserved
13	PS2_ACT_0	N	PS2_0 Activity Interrupt form PS/2 Block
14	PS2_ACT_1	N	PS2_1 Activity Interrupt form PS/2 Block
15	PS2_ACT_2	N	PS2_2 Activity Interrupt form PS/2 Block
16	Reserved	N	Reserved
17	PS2_WK_0A	Y	PS2_0A Start Detectform pin signal
18	PS2_WK_0B	Y	PS2_0B Start Detectform pin signal
19	PS2_WK_1A	Y	PS2_1A Start Detectform pin signal
20	PS2_WK_1B	Y	PS2_1B Start Detectform pin signal

TABLE 16-54: BIT DEFINITION GIRQ19 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
21	PS2_WK_2	Y	PS2_2 Start Detectform pin signal
30-22	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-9 Source bits have corresponding Enable and Result bits.

16.6.40 GIRQ19 ENABLE REGISTER

TABLE 16-55: GIRQ19 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	E0h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-54 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-54, “Bit Definition GIRQ19 Source Register,” on page 281.						

16.6.41 GIRQ19 RESULT REGISTER

TABLE 16-56: GIRQ19 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	E4h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-54, “Bit Definition GIRQ19 Source Register,” on page 281.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.42 GIRQ20 SOURCE REGISTER

TABLE 16-57: GIRQ20 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	F0h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-58 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-58, “Bit Definition IRQ20 Source Register,” on page 283.						

TABLE 16-58: BIT DEFINITION IRQ20 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	EC2_OBF	N	Embedded Controller 2 OBF Interrupt. This interrupt is signaled when the OBF bit in the EC2 status register has been clear.
1	EC2_IBF	N	Embedded Controller 2 IBF Interrupt. This interrupt is signaled when the host writes to the EC2 Command or Data port.
2	EC1_OBF	N	Embedded Controller 1 OBF Interrupt. This interrupt is signaled when the OBF bit in the EC1 status register has been clear.
3	EC1_IBF	N	Embedded Controller 1 IBF Interrupt. This interrupt is signaled when the host writes to the EC1 Command or Data port.
4	EC0_OBF	N	Embedded Controller 0 OBF Interrupt. This interrupt is signaled when the OBF bit in the EC0 status register has been clear.
5	EC0_IBF	N	Embedded Controller 0 IBF Interrupt. This interrupt is signaled when the host writes to the EC0 Command or Data port.
6	EC3_OBF	N	Embedded Controller 3OBF Interrupt. This interrupt is signaled when the OBF bit in the EC3 status register has been clear.
7	EC3_IBF	N	Embedded Controller 3 IBF Interrupt. This interrupt is signaled when the host writes to the EC3 Command or Data port.
30-8	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-10 Source bits have corresponding Enable and Result bits.

16.6.43 GIRQ20 ENABLE REGISTER

TABLE 16-59: GIRQ20 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	F4h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Refer to Table 16-58, “Bit Definition IRQ20 Source Register,” on page 283.						

16.6.44 GIRQ20 RESULT REGISTER

TABLE 16-60: GIRQ20 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	C8h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 16-58, “Bit Definition IRQ20 Source Register,” on page 283.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.45 GIRQ21 SOURCE REGISTER

TABLE 16-61: GIRQ21 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	104h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 16-62 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-62, “Bit Definition IRQ21 Source Register,” on page 285.						

TABLE 16-62: BIT DEFINITION IRQ21 SOURCE REGISTER

Bit	Signal Name	Wake	Description
4:0	VLPC_EVT[0:4]	N	VLPC Companion 0, events 0:4 Interrupts
9:5	VLPC_EVT[5:9]	N	VLPC Companion 1, events 0:4 Interrupts
14:10	VLPC_EVT[10:14]	N	VLPC Companion 2, events 0:4 Interrupts
15	VLPC_TPM	N	Interrupt from TPM on VLPC
16	VLPC_EC	N	VLPC controller interrupt
17	V_DATA	Y	VLPC bus wakeup event (attention signal from Companion)
30:18	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 16-11 Source bits have corresponding Enable and Result bits.

16.6.46 GIRQ21 ENABLE REGISTER

TABLE 16-63: GIRQ21 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	108h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 16-62 & Note 16-1 on page 265.						
BIT NAME	Reserved	Refer To Table 16-62, “Bit Definition IRQ21 Source Register,” on page 285.						

16.6.47 GIRQ21 RESULT REGISTER

TABLE 16-64: GIRQ21 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	10Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer To Table 16-62, “Bit Definition IRQ21 Source Register,” on page 285.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.48 GIRQ22 SOURCE REGISTER

TABLE 16-65: GIRQ22 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	118h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	Reserved	Refer To Table 16-66, “Bit Definitions GIRQ22 Source Register,” on page 286.						

TABLE 16-66: BIT DEFINITIONS GIRQ22 SOURCE REGISTER

Bit	Signal Name	Wake	Description
[15:0]	GPIO	Y	GPIO[217:200]
[30:16]	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

16.6.49 GIRQ22 ENABLE REGISTER

TABLE 16-67: GIRQ22 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	11Ch					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Refer To Table 16-66, “Bit Definitions GIRQ22 Source Register,” on page 286.						

16.6.50 GIRQ22 RESULT REGISTER

TABLE 16-68: GIRQ22 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	120h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer To Table 16-66, “Bit Definitions GIRQ22 Source Register,” on page 286.						

NORM

This bit is a read write bit used by the NORM instruction.

16.6.51 GIRQ23 SOURCE REGISTER

TABLE 16-69: GIRQ23 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	12Ch					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	Reserved	Refer To Table 16-70, “Bit Definitions GIRQ23 Source Register,” on page 287.						

TABLE 16-70: BIT DEFINITIONS GIRQ23 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	TIMER0	N	16-bit Timer Interrupt
1	TIMER1	N	16-bit Timer Interrupt
2	TIMER2	N	16-bit Timer Interrupt
3	TIMER3	N	16-bit Timer Interrupt
4	Reserved	N	Reserved
5	PFR	N	Power Fail Register Interrupt
6	Reserved	N	Reserved
7	WEEK_ALR	Y	Week Alarm Interrupt. Week Timer has reached it’s terminal count.
8	VCI_OVRD_IN	Y	Pin input of VCI_OVRD_IN
9	VCI_IN0	Y	input of VCI_IN0# pins
10	VCI_IN1	Y	input of VCI_IN1# pins
11	VCI_IN2	Y	input of VCI_IN2# pins
12	VCI_IN3	Y	input of VCI_IN3# pins

TABLE 16-70: BIT DEFINITIONS GIRQ23 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
13	HTIMER0	Y	Hibernation Timer Interrupt. This interrupt is signaled when the hibernation timer has counter down to zero.
14	HTIMER1	Y	Hibernation Timer Interrupt. This interrupt is signaled when the hibernation timer has counter down to zero.
15	Reserved	N	Reserved
16	CAPTURE TIMER	N	The Free Running timer in the Capture/Compare Time transitioned from FFFF_FFFFh to 0000_0000h
17	CAPTURE 0	N	Capture register 0 in the Capture/Compare Timer unit has acquired a new value
18	CAPTURE 1	N	Capture register 1in the Capture/Compare Timer unit has acquired a new value
19	CAPTURE 2	N	Capture register 2 in the Capture/Compare Timer unit has acquired a new value
20	CAPTURE 3	N	Capture register 3 in the Capture/Compare Timer unit has acquired a new value
21	CAPTURE 4	N	Capture register 4 in the Capture/Compare Timer unit has acquired a new value
22	CAPTURE 5	N	Capture register 5 in the Capture/Compare Timer unit has acquired a new value
23	COMPARE 0	N	Compare register 0 in the Capture/Compare Time unit tripped
24	COMPARE 1	N	Compare register 1 in the Capture/Compare Time unit tripped
[30:25]	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

16.6.52 GIRQ23 ENABLE REGISTER

TABLE 16-71: GIRQ23 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	130h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Refer To Table 16-70, “Bit Definitions GIRQ23 Source Register,” on page 287.						

16.6.53 GIRQ23 RESULT REGISTER

TABLE 16-72: GIRQ23 RESULT REGISTER

HOST ADDRESS							**HOST SIZE**	
EC OFFSET	134h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W-	R	R	R	R	R	R	R
BIT NAME	NORM	Refer To Table 16-70, “Bit Definitions GIRQ23 Source Register,” on page 287.						

BIT 31 NORM

This bit is a read write bit used by the NORM instruction.

16.6.54 IA GROUP SELECT REGISTER

TABLE 16-73: IA GROUP SELECT REGISTER

HOST ADDRESS							**HOST SIZE**	
EC OFFSET	200h					32-bit	**EC SIZE**	
POWER	VTR					00FF_FF00h	**VTR POR DEFAULT**	
BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	GIRQ_Enable[23:16]							
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	GIRQ_Enable[15:8]							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

GIRQ_ENABLE[23:8]

Group IRQ Enable. Enable or disable all interrupts in a group.

0= All Interrupts in the associated GIRQ will be disabled.

1= All Interrupts in the associated GIRQ will be enabled.

16.6.55 IRQ ENABLE REGISTER

TABLE 16-74: IRQ ENABLE REGISTER

HOST ADDRESS							**HOST SIZE**	
EC OFFSET	204h					32-bit	**EC SIZE**	
POWER	VTR					00FF_FF00h	**VTR POR DEFAULT**	
BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	IRQ Enable[23:16]							
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	IRQ Enable[15:8]							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R	R	R
BIT NAME	IRQ Enable[7:3]					Reserved		

IRQ ENABLE[31:0]

IRQ Enable. Enable or disable all IRQs going to the ARC.

0= Respective IA IRQi will be disabled.

1= Respective IA IRQi will be enabled.

16.7 Extension Core Registers

16.7.1 EXTENSION CORE REGISTER R56

Register R56 is the Interrupt Aggregator Vector register. The Interrupt Aggregator automatically generates an address of a 4-byte location in the EC address space. The location contains a 4-byte address of an interrupt handler for the highest priority interrupt selected by the Interrupt Aggregator.

This register can be used in any ARC instruction with a 6-bit register address field.

TABLE 16-75: IA VECTOR REGISTER R56

BIT	D31	D30	D29	D28	D27	D26	D25	D24
TYPE	R	R	R	R	R	R	R	R
FIELD	0	0	0	0	0	0	0	0
BIT	D23	D22	D21	D20	D19	D18	D17	D16
TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
FIELD	VT_Base[12:5]							
BIT	D15	D14	D13	D12	D11	D10	D9	D8
TYPE	R/W	R/W	R/W	R/W	R/W	R	R	R
FIELD	VT_Base[4:0]					Group_Select[3:1]		
BIT	D7	D6	D5	D4	D3	D2	D1	D0
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	Group_ Select[0]	Bit_Select[4:0]					0	0

D[6:2]: BIT_SELECT[4:0]

This field is the index of the interrupt with the highest priority within the current group that is asserted, enabled, and set to a priority level that is equal to or greater than the current IA priority level. The interrupt with the highest bit number has the highest priority within a group.

This field is set by hardware and is read-only.

D[10:7]: GROUP_SELECT[3:0]

This field is the index of the GIRQ with the highest priority in the Interrupt Aggregator among GIRQs in which at least one interrupt is asserted, enabled and set to a priority level that is equal to or greater than the current IA priority level. The GIRQ with the highest index has the highest priority among the GIRQs.

This field is set by hardware and is read-only.

VT_BASE[11:0]

This field corresponds to bits 23:11 of the Interrupt Vector Table in the AHB address space. The IQ Vector Register is an address in the 24-bit AHB address space which contains the address of an interrupt handler. The AHB address space includes the Boot Memory, the Flash Memory and the Data/Instruction SRAM.

This field can be read and written by firmware. It should be initialized to the base address of the Interrupt Vector Table before interrupts are enabled.

16.7.2 EXTENSION CORE REGISTER R55

Extension Core Register R55 is a 6-bit register that sets the current priority level for the Interrupt Aggregator. Bits 31:6 are reserved and always return 0 on reads. An interrupt priority level of 4 or higher disables all interrupts.

This register can be used in any ARC instruction with a 6-bit register address field.

TABLE 16-76: IA PRIORITY REGISTER R55

BIT	D31	D30	D29	D28	D27	D26	D25	D24
TYPE	R	R	R	R	R	R	R	R
FIELD	Reserved							
BIT	D23	D22	D21	D20	D19	D18	D17	D16
TYPE	R	R	R	R	R	R	R	R
FIELD	Reserved							
BIT	D15	D14	D13	D12	D11	D10	D9	D8
TYPE	R	R	R	R	R	R	R	R
FIELD	Reserved							
BIT	D7	D6	D5	D4	D3	D2	D1	D0
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved		Hi_Pri			Current_Priority		

CURRENT_PRIORITY[2:0]

This is a read/write field that determines the current priority level for the Interrupt Aggregator. Only interrupts that are configured to be at this priority level or higher will be used to generate the interrupt vector. A Current_Priority value of 4 or higher will block all interrupts in the Interrupt Aggregator.

HI_PRI

This three-bit field is set whenever Extension Core Register R56 (IA Vector Register) is read. The value is the **priority level + 1** assigned to the interrupt that corresponds to the vector read in R56. It is the hardware that adds the 1 to the current priority level of the generated interrupt. For example, if the generated interrupt had a priority 2 then hardware will add 1 to it and write a value of 3 to the Hi_Pri bits.

17.0 WATCHDOG TIMER INTERFACE

17.1 General Description

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

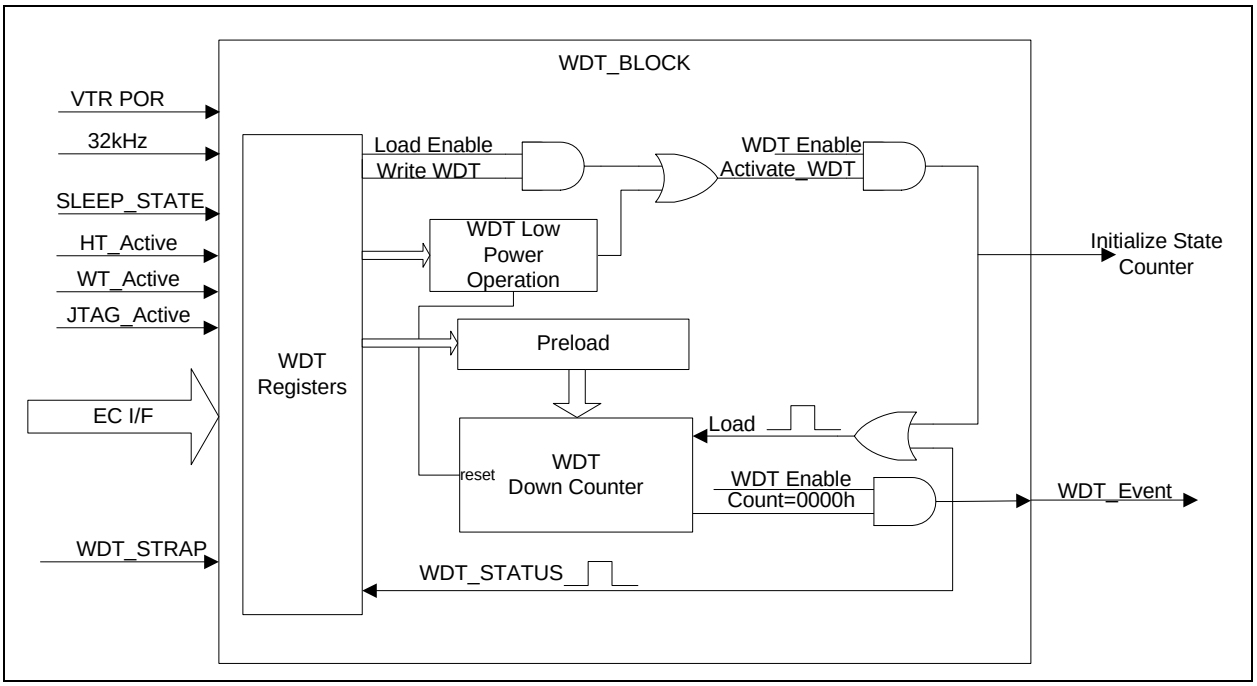
When enabled, the WATCHDOG Timer (WDT) circuit will generate a WDT Event if the user program fails to reload the WDT within a specified length of time known as the WDT Interval.

This timer can be held inactive via the WDT Stall feature if the Hibernation timer, Week Timer, or the JTAG interface are enabled and active. This featured if enabled can be used to avoid unintended system resets.

Some operations can be carried out without any delay, e.g., registers can be read at any time and disabling the WDT takes effect immediately. On the other hand, 'kicking' the WDT may have a latency of up to 1 32-kHz cycle (~ 30 us). Similarly, when the load register is altered, the WDT cannot be enabled for up to 1 32-kHz cycle. Note that the ring oscillator must not be stopped within one 32-kHz clock following register write events.

17.2 Block Diagram

FIGURE 17-1: Watchdog Timer Interface BLOCK DIAGRAM



17.3 Watchdog Timer Interface Signal List

TABLE 17-1: Watchdog Timer Interface SIGNAL LIST

Signal Name	Direction	Description
nSYS_RST	INPUT	Power on Reset to the block
32kHz	INPUT	X32K_CLK, Clock source for WDT logic
HT_Active	INPUT	Signal indicating the Hibernation Timer is active and counting. See Section 20.0, "Hibernation Timer," on page 320.
WT_Active	INPUT	Signal indicating the Week Timer is active and counting. See Section 21.0, "Week Alarm Interface," on page 324.
JTAG_Active	INPUT	Signal indicating the JTAG interface is active. See Section 39.0, "JTAG and XNOR," on page 477.

TABLE 17-1: Watchdog Timer Interface SIGNAL LIST (CONTINUED)

Signal Name	Direction	Description
SPB Interface	I/O Bus	Bus used by microprocessor to access the registers in this block.
WDT Event	OUTPUT	Pulse generated when WDT expires (Note 17-1)

Note 17-1 In the MEC1609/MEC1609i, the WDT Event output is routed to the WDT_ALRT input (see Table 5-1, “Power, Clocks and Resets Port List,” on page 74). Asserting the WDT Event output causes a Watch-Dog Timer Forced Reset. (See Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99.)

17.4 Power, Clocks and Reset

17.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

17.4.2 CLOCKS

This block has two clock inputs, the EC Bus Clock and X32K_CLK. The EC Bus Clock is used in the interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK is the clock source for the Watchdog Timer functional logic, including the counter.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

17.4.3 RESET

This block is reset on a nSYS_RST.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

17.5 WDT Event Output Routing

The WDT Event (output) causes the Section 5.6.10, "Watch-Dog Timer Forced Reset," on page 99.

The WDT Event state is also retained through a Watch-Dog Timer Forced Reset in the WDT bit of the Power-Fail and Reset Status Register on page 111. The Power-Fail and Reset Status Register can generate a interrupt via the PFR interrupt GIRQ23 Source Register on page 287.

The WDT Event output is not directly connected to an EC interrupt.

17.6 WDT Operation

17.6.1 WDT ACTIVATION MECHANISM

The WDT is activated by the following sequence of operations during normal operation. Note that the WDT Load Register can be programmed only when WDT is disabled (WDT Enable = ‘1’).

1. Load the WDT Load Register with the count value. ‘0’ is an invalid load value.
2. Set the WDT Enable bit in the WDT Control Register.

The WDT Activation Mechanism starts the WDT decrementing counter.

17.6.2 WDT DEACTIVATION MECHANISM

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

17.6.3 WDT RELOAD MECHANISM

The WDT must be reloaded within periods that are shorter than the programmed watchdog interval; otherwise the WDT will underflow and a WDT Event will be generated and the WDT Status bit will be set in the WDT Control Register. It is the responsibility of the user program to continually execute sections of code which reload the watchdog timer (WDT) causing the counter to be reloaded.

There are two methods of reloading the WDT: a write to the WDT Kick Register or the WDT Activation Mechanism.

17.6.4 WDT INTERVAL

The WDT Interval is the time it takes for the WDT to decrements from the WDT Load Register value to 0000h. The WDT Count Register value takes 1.007ms to decrement by 1 count.

17.6.5 WDT STALL OPERATION

The WDT has several events that can cause the WDT STALL. When a WDT STALL event is asserted, the WDT stops decrementing, and the WDT enters a low power state. When a WDT STALL event is de-asserted, the counter resumes decrementing from the count at which it stopped.

The WDT STALL feature has been implemented for convenience. If the system designer chooses not to utilize the WDT STALL feature, the WDT defaults with the WDT STALL feature disabled.

There are three Stall inputs to the WDT: HT_Active, WT_Active, and JTAG_Active, corresponding to the Hibernation Timer, the Week Alarm Timer, & the J-TAG interface being active. The Stall inputs have individual enable bits: HT STALL_EN, WT STALL_EN, JTAG STALL_EN bits in the WDT Control Register on page 297.

Note 17-2 Only a single instance of the Hibernation Timer (Hibernation Timer.0 on page 321) is routed to the HT_Active stall input of the Watchdog Timer Interface.

TABLE 17-2: WDT STALL EVENT BEHAVIOR

WDT Stall Input (Activity Indicator)	WDT Control Register on page 297		WDT Behavior	WDT Event Output
	STALL_EN BIt	WDT Enable Bit		
X	X	0	Counter is reset and not active. Clock source to counter is gated to save power.	0
X	0	1	Count is active. If counter > 0000h	1
0	1	1		
X	X	1	Count is decremented to 0000h	1
1	1	1	Counter is not active. Clock source to counter is gated to save power.	0

Note 17-3 When the counter reaches 0000h it wraps to the preload value and starts counting down again. This creates a pulse on the WDT Event output.

17.7 Instance Description

There is one instance of the Watchdog Timer Interface block implemented in the MEC1609/MEC1609i.

Each instance of the Watchdog Timer Interface has its own Logical Device Number, and Base Address as indicated in Table 17-3.

TABLE 17-3: Watchdog Timer Interface BASE ADDRESS TABLE

Watchdog Timer Instance	LDN from (Table 3-2 on page 48)	AHB Base Address
Watchdog Timer	1h	F0_0400h

The Table 17-4 is a register summary for one instance of the Watchdog Timer Interface. Each EC address is indicated as an SPB Offset from its AHB address.

TABLE 17-4: Watchdog Timer Interface REGISTER SUMMARY

Register Name	EC Interface			Notes
	SPB Offset	Byte Lane	EC Type	
WDT Load Register	00h	0	R/W	
		1		
WDT Control Register	04h	0	R/W	
WDT Kick Register	08h	0	W	
WDT Count Register	0Ch	0	R	
		1		

Note: All Registers listed in Table 17-4 are powered by VTR and reset by nSYS_RST.

Note 17-4 All register are clocked by the EC Bus Clock.

17.8 Detailed Register Descriptions

17.8.1 WDT LOAD REGISTER

TABLE 17-5: WDT LOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	WDT load[15:0]							

WDT LOAD[15:0]

Writing this field reloads the Watch Dog Timer counter.

WDT Load Register can be programmed only when WDT Enable = '0'.

'0' is not a valid load value.

To verify that load has taken place, it is recommended that software polls the WDT Count Register until its value reflects that of the new load value.

17.8.2 WDT CONTROL REGISTER

TABLE 17-6: WDT CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VTR			00h (Note 17-5)			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/WC	R/W
BIT NAME	Reserved			JTAG STALL_EN	WT STALL_EN	HT STALL_EN	WDT Status	WDT Enable

Note 17-5 The default for the WDT Control Register changes depending on the state of the WDT Status bit.

WDT ENABLE

Note:	The default of the WDT is inactive.

In WDT Operation, the WDT is activated by the sequence of operations defined in Section 17.6.1, "WDT Activation Mechanism" and deactivated by the sequence of operations defined in Section 17.6.2, "WDT Deactivation Mechanism". In WDT STALL Operation, hardware may be enabled to automatically activate and deactivate the WDT.

WDT STATUS

WDT Status is set by hardware if the last reset of MEC1609/MEC1609i was caused by an underflow of the WDT. See Section 17.6.3, "WDT Reload Mechanism," on page 294 for more information.

This bit must be cleared by the EC firmware writing a ‘1’ to this bit. Writing a ‘0’ to this bit has no effect.

JTAG STALL_EN

This bit is used to enable the JTAG_Active (JTAG_RST# pin not asserted) WDT STALL Operation on page 295.

0= JTAG_Active WDT STALL Operation not enabled

1= JTAG_Active WDT STALL Operation enabled

WT STALL_EN

This bit is used to enable the WT_Active (Week Timer) WDT STALL Operation on page 295.

0= WT_Active WDT STALL Operation events not enabled

1= WT_Active WDT STALL Operation events enabled

HT STALL_EN

This bit is used to enable the HT_Active (Hibernation Timer) WDT STALL Operation on page 295.

0= HT_Active WDT STALL Operation events not enabled

1= HT_Active WDT STALL Operation events enabled

17.8.3 WDT KICK REGISTER

TABLE 17-7: WDT KICK REGISTER

HOST ADDRESS	n/a				n/a			**HOST SIZE**
EC OFFSET	08h				8-bit			**EC SIZE**
POWER	VTR				n/a			nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	Kick							

KICK

The WDT Kick Register is a strobe. Reads of the WDT Kick Register return 0.

Writes to the WDT Kick Register cause the WDT to reload the WDT Load Register value and start decrementing when the WDT Enable bit in the WDT Control Register is set to '1'. When the WDT Enable bit in the WDT Control Register is cleared to '0', writes to the WDT Kick Register have no effect.

17.8.4 WDT COUNT REGISTER

TABLE 17-8: WDT COUNT REGISTER

HOST ADDRESS	n/a				n/a			**HOST SIZE**
EC OFFSET	0Ch				16-bit			**EC SIZE**
POWER	VTR				FFFFh			nSYS_RST **DEFAULT**
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	WDT COUNT[15:0]							

WDT COUNT[15:0]

This read-only register provide the current WDT count.

18.0 EC AHB SPI FLASH READ CONTROLLER

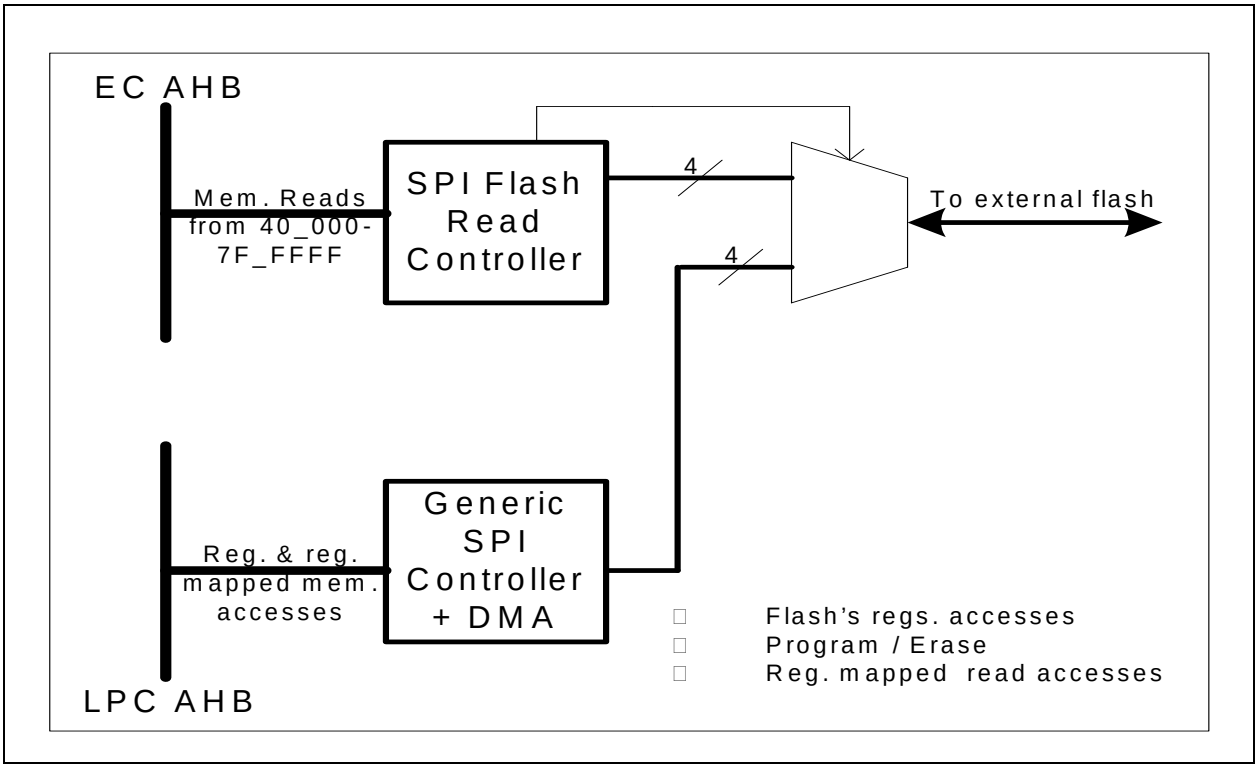
18.1 Overview

Two SPI master modules serve as interface to the external serial flash. The SPI controller on LPC AHB, General Purpose Serial Peripheral Interface (GP-SPI) described in Section 31.0, provides flash register programming interface as well as erase/write. The SPI controller on EC AHB bus described in this chapter is solely used for reading from flash and is optimized for code fetch from the EC cache controller.

SPI signals to/from the two modules are multiplexed at the top level. The two controllers operate independently and are not aware of the flash's status as a result of transactions initiated by the other. It is software's responsibility to manage the switch. Status register bits are available from both controllers; register bit that controls the multiplexing is in the AHB SPI Flash Read controller.

The AHB SPI Read Controller is addressed as EC Logical Device 9h with base address F0_2800h.

FIGURE 18-1: EXTERNAL SPI FLASH INTERFACES



18.2 Features

- SPI Mode 0 flash (output shifted out on SPI clock falling edge; input shifted in on rising edge)
- SPI clock up to 64 MHz (64-, 32-, and 16-MHz)
- Support read, fast read, and fast read with dual output opcodes
- Up to 4 MB in size
- AHB burst accesses will be SPLIT and controller will provide at most 1 word at a time.
- Memory interface optimized for 16-byte reads (cache line fill)
- Programmable memory chip select de-assertion time
- Programmable memory chip select assertion time post transaction (to avoid command and opcode overhead in case of consecutive sequential accesses)

18.3 Addressing

The EC address range 40_0000h-7F_FFFFh is reserved for external flash. This area is cacheable instruction space; the cache is a 2-KB direct mapped cache. Read accesses within this range are forwarded to the AHB SPI flash read controller. On the memory side, the address shifted into the SPI flash can be either the AHB address or a mapped version that is formed by concatenating the contents of the SPI Page Address Register, which forms the high part of the address, with lower AHB address lines. A register in the AHB bus controller specifies how many AHB address lines are to be used.

18.4 Operation

18.4.1 SWITCHING BETWEEN FLASH READ CONTROLLER AND FW-DIRECT SPI CONTROLLER

A signal from the AHB Flash Read Controller controls the multiplexing of SPI signals from the two modules.

When the Master Bridge Enable bit in the Control and Status Register is set, the multiplexer immediately selects SPI signals from the AHB SPI Controller. Therefore, ownership transfer in this direction requires a software protocol between the drivers or driver and OS to ensure the flash memory is in idle state before switching.

To switch control back to the FW-Direct SPI Controller, software clears Master Bridge Enable and then polls the Bridge Using SPI Flash status bit. Control is switched when the latter is ‘0’.

18.4.2 PROGRAMMING SET UP

If paged access is desired, software needs to program appropriate values into the SPI Page Address Register and Number_of_AHB_addresses registers. See Section 18.3. In the MEC1609/MEC1609i, these registers default respectively to 01h and 15h, giving rise to SPI address within 4x_xxxxh to 7x_xxxxh.

Next, memory-dependent parameters such as SPI clock frequency and chip enable de-assertion time are specified. Supported frequencies include 64-, 32-, and 16-MHz. Time is expressed in terms of 64-MHz cycles. Fast_Read opcode should be used for high-speed operation; currently, this usually means clock speed of 20 MHz or higher and insertion of Dummy Byte is required. Options related to performance and power consumption are described separately in Section 18.4.4 and Section 18.4.5, respectively.

The controller is enabled by setting the Master Bridge Enable bit in the Control and Status Register register.

18.4.3 DATA TRANSFER

Once set up and enabled, the controller responds to memory reads within address range 40_0000h-7F_FFFFh. Data are fetched from memory and then provided to the requestor one word at a time. The initial response to any such transactions is always a SPLIT (AHB bus terminology - retry in English). The controller latches the AHB bus master’s ID, bus address, and requested transfer size (only up to 16 bytes is recorded). It then initiates SPI read cycles using programmed parameters described in Section 18.4.2. After the first word has been fetched into the internal buffer, the controller informs the AHB bus arbiter to re-grant access to the aforementioned master. The latter restarted the same transaction and receives up to one word of data. For multiple-word transactions, the controller resume fetching data from memory into the newly available internal buffer while again SPLITting the AHB transaction. This process continues until the last word has been provided to the AHB master.

The controller serves only one AHB master at a time. Attempts to read from SPI memory by other masters will be SPLITted. Masters’ IDs are recorded but cycles’ information are not. Upon transferring the last data word to the active master, the controller informs the bus arbiter to regrant bus access to these pending masters.

18.4.4 PERFORMANCE CONSIDERATIONS

There are various options to increase performance, i.e., to reduce the time it takes to provide read data to AHB master.

Chip enable can be kept active after all requested data have been fetched: If the next read access is sequential, there is no need to present the flash memory with opcode and address. As long as chip enable remains asserted, the memory outputs bits in the next sequential word with each new clock pulse. This represents a saving of 8 (opcode) + 24 (address) SPI clock cycles. On the other hand, if the chip enable is kept asserted but the next transaction is not sequential, it has to be de-asserted for a minimum time specified in CE De-assertion Time Register before it can be asserted again for the new cycle. A typical value for this penalty is between 2-7 SPI clock cycles.

The delay of CE assertion to first SPI clock can be reduced from 2 to 1 64-MHz clock by setting the CE Setup Speed-up bit in the Control and Status Register. This feature can be enable always.

Under certain condition, the SPI PKT Start Speed-up bit can be set to shorten the initiation of next SPI packet fetch by (AHB clock divisor - 1) 64-MHz clock.

18.4.5 POWER CONSUMPTION CONSIDERATION

Power consumption is about 3 orders of magnitude lower when the memory is not selected. The controller can be set up to de-assert the chip enable signal immediately after a transaction. This is done by writing 0h to the CE Extended Active Time Register.

18.5 AHB SPI Read Controller Registers

The controller is assigned EC LDN 9h; its registers have base address F0_2800h.

TABLE 18-1: AHB SPI FLASH READ CONTROLLER REGISTERS

Register Name	EC Offset	Access Type	VTR POR Default
Control and Status Register	00h	R/W-RO	01h
SPI Command Opcode Register	04h	R/W	03h
CE De-assertion Time Register	08h	R/W	07h
CE Extended Active Time Register	0Ch	R/W	0010h
SPI Page Address Register	10h	R/W	00h

TABLE 18-2: CONTROL AND STATUS REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 18-1 on page 301					32-bit	**EC SIZE**	
POWER	VTR			See Table 18-1 on page 301			**VTR POR DEFAULT**	
	EC SPB							
BYTES[3:1]	**D31**	**D30**	**. . .**				**D9**	**D8**
HOST TYPE	-		. . .				-	-
EC TYPE	R	R	. . .				R	R
BYTE[1] BITS	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-		-		-		-	-
EC TYPE	R	R/W	R/W	R/W	R/W		RO	R/W
BIT NAME	SPI PKT Start Speed-up	CE Setup Speed-up	Fast Read Dual Output Mode	Dummy Byte Insertion	SPI Clock Frequency		Bridge Using SPI Flash	Master Bridge Enable

MASTER BRIDGE ENABLE

Set to enable the AHB SPI flash read controller. This bit also controls the multiplexing of SPI signals at the top level. See Switching between Flash Read Controller and FW-Direct SPI Controller on page 300.

BRIDGE USING SPI FLASH

When set, this status bit indicates AHB SPI flash read controller is in control of the SPI flash interface.

SPI CLOCK FREQUENCY

The SPI clock is derived from the 64-MHz master clock as follows:

‘00’: SPI clock = master clock divided by 4

‘01’: SPI clock = master clock divided by 2

‘10’: SPI clock = master clock divided by 1 but inverted

‘11’: Reserved

DUMMY BYTE INSERTION

Set to enable the insertion of a dummy byte after the 3-byte flash address. This is normally required with Fast Read opcode.

FAST READ DUAL OUTPUT MODE

Set to configure the controller for fast dual read mode in which data are retrieved from memory on both DIO and DO pins, i.e., even bits on DIO and odd bits on DO. This bit controls the demultiplexing of input data. Note that the SPI Command Opcode must be set accordingly, and, as with Fast Read, dummy byte insertion is normally required.

CE SETUP SPEED-UP

Set to reduce the chip enable active set up time from 2 to 1 64-MHz clock.

SPI PKT START SPEED-UP

Set to speed up the initiation of the next data fetch from SPI memory. This is allowed when the AHB bus clock is x times SPI clock. The initiation is shortened by (AHB clock divisor - 1) master clock.

TABLE 18-3: SPI COMMAND OPCODE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 18-1 on page 301					32-bit	**EC SIZE**	
POWER	VTR					See Table 18-1 on page 301	**VTR POR DEFAULT**	
	EC SPB							
BYTES[3:1]	**D31**	**D30**	**. . .**				**D9**	**D8**
HOST TYPE	-		. . .				-	-
EC TYPE	R	R	. . .				R	R
BYTE[1] BITS	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-							
EC TYPE	R/W							
BIT NAME	SPI Command Opcode							

SPI COMMAND OPCODE

Read opcode sent to memory during command phase. Valid opcodes are:

03h: Read

0Bh: Fast Read

3Bh: Fast Read with dual output. See the description of the Control and Status Register.

TABLE 18-4: CE DE-ASSERTION TIME REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 18-1 on page 301					32-bit	**EC SIZE**	
POWER	VTR			See Table 18-1 on page 301			**VTR POR DEFAULT**	
	EC SPB							
BYTES[3:1]	**D31**	**D30**	**. . .**				**D9**	**D8**
HOST TYPE	-		. . .				-	-
EC TYPE	R	R	. . .				R	R
BYTE[1] BITS	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-							
EC TYPE	R/W							
BIT NAME	CE De-assertion Time							

CE DE-ASSERTION TIME

This must be equal to or greater than the minimum CE de-assertion time specified by the memory manufacturer. The programmed value is expressed in units of 64-MHz cycles with the de-assertion time equal to the number of cycles + 1.

TABLE 18-5: CE EXTENDED ACTIVE TIME REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 18-1 on page 301					32-bit	**EC SIZE**	
POWER	VTR			See Table 18-1 on page 301			**VTR POR DEFAULT**	
	EC SPB							
BYTE[3:2]	**D31**	**D30**	**. . .**				**D17**	**D16**
HOST TYPE	-		. . .				-	-
EC TYPE	R	R	. . .				R	R
BYTE[1:0]	**D15**	**D14**					**D1**	**D0**
HOST TYPE	-							
EC TYPE	R/W							
BIT NAME	CE Extended Active time							

CE EXTENDED ACTIVE TIME

Specified in units of 64-MHz clock cycles; the extended time is equal to the number of cycle + 2. If non-zero, the memory chip enable signal is kept asserted for the specified time after the requested data have been fetched. If the next transaction is a read to the next sequential address, the controller needs not provide command and address to the memory again. The next sequential data are fetched simply by toggling SPI clock.

When CE is being extended and a new (and different) value is programmed into the register, CE will be either shortened or extended accordingly.

TABLE 18-6: SPI PAGE ADDRESS REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 18-1 on page 301					32-bit	**EC SIZE**	
POWER	VTR				See Table 18-1 on page 301		**VTR POR DEFAULT**	
	EC SPB							
BYTE[3:1] BITS	**D31**	**D30**	**. . .**				**D9**	**D8**
HOST TYPE	-		. . .				-	-
EC TYPE	R	R	. . .R				R	R
BYTE1 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-							
EC TYPE	R/W							
BIT NAME	Page Address							

PAGE ADDRESS

If not all 24-bit AHB address lines are used to form the SPI address, the Page Address bits are used to form the upper part of the SPI address. The lower part is taken from the AHB bus address; the number of address lines of the latter that are used is specified in a register in the AHB bus controller.

18.6 Sleep Interface

The EC AHB SPI Flash Read Controller clock required output is asserted following nSYS_RST. If the Master Bridge Enable bit is asserted and the EC AHB SPI Flash Read Controller sleep enable input is not asserted, the clock required output remains asserted.

If the Master Bridge Enable is de-asserted, or the sleep enable input is asserted, the clock required output remains asserted while the EC AHB SPI Flash Read Controller is ‘busy.’

If the Master Bridge Enable is de-asserted, or the sleep enable input is asserted, the clock required output is de-asserted when the EC AHB SPI Flash Read Controller is ‘idle’ and remains not asserted until the Master Bridge Enable bit is asserted and the sleep enable input is de-asserted.

Note that except for the clock required status bit there are no registered bits within the EC AHB SPI Flash Read Controller to indicate the ‘idle’ state.

19.0 16-BIT TIMER INTERFACE

19.1 General Description

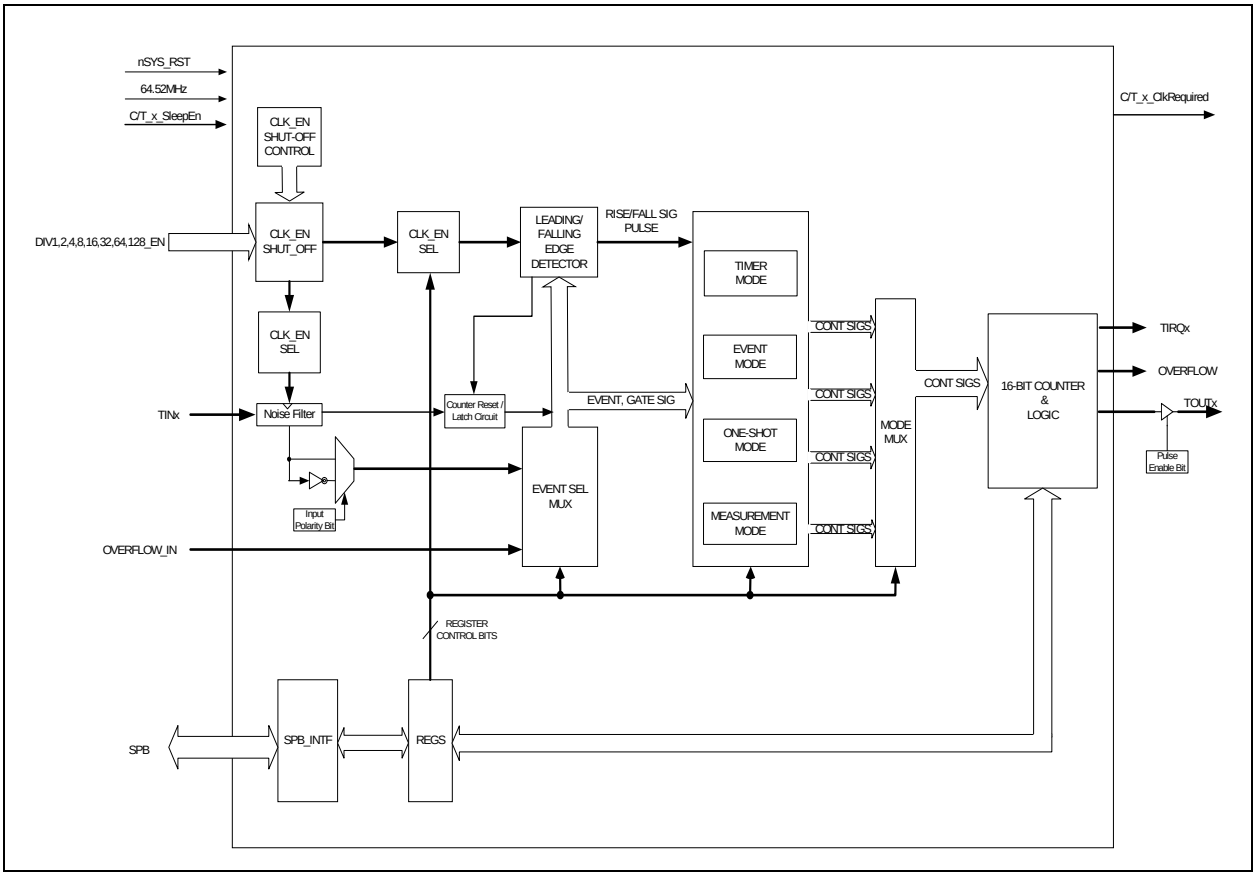
The MEC1609/MEC1609i 16-Bit Timer Interface implements four 16-bit auto-reloading timer/counters. Each timer/counter is categorized as one of three types: General Purpose, Input-Only and Input/Output. All timer/counters have four modes of operation: Timer, One-Shot, Event and Measurement. In addition, each timer/counter can generate a unique wake-up interrupt to the EC. The clock for each timer/counter is derived from the system clock and can be divided down by a prescaler. Input-Only and Input/Output timers can also use an external input pin to clock or gate the counter. To aid operation in noisy environments the external input pin also has a selectable noise filter. If large counts are required, the output of each timer/counter can be internally connected to the next timer/counter.

The following section defines terms used in this chapter.

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

19.2 Block Diagram

FIGURE 19-1: BLOCK DIAGRAM FOR TIMER X



19.3 Signal List for Block Diagram

TABLE 19-1: BLOCK DIAGRAM SIGNAL LIST DESCRIPTION

Signal Name	Direction	Description
VTR POR	INPUT	nSYS_RST
64.52MHz_CLK	INPUT	Clock source to block.
DIV1,2,4,8,16,32,64,128_EN	INPUT	Clock Enables for supporting Filter and Timer frequencies.
TINx	INPUT	Timer x Input signal
TIRQx	OUTPUT	Timer x Interrupt Request
C/T_x_SleepEn, x=0-3	INPUT	Sleep Enable signals to counters 1-4
C/T_x_ClkRequired, x=0-3	INPUT	Clock required signals from counters 1-4
TOUTx	OUTPUT	Timer x Output signal
SPB_IF	I/O Bus	Bus used by microprocessor to access the registers in this block.

19.4 Timer Connections

For external inputs/outputs (TINx/TOUTx) to/from timers, please see Section 2.4.16, "16-Bit Counter/Timer Interface," on page 20.

TABLE 19-2: TIMER CASCADING DESCRIPTION

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

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

19.5 Power, Clocks and Reset

19.5.1 POWER DOMAIN

This block is powered by the VTR power supply.

19.5.2 CLOCKS

There is a clock enable input for each of the supported frequencies listed in Table 19-12, "Timer Clock Frequencies," on page 317. Any of these enables may be selected for the Timer Clock Frequency. Independently, any of these clock frequencies may be selected for the filter clock via the FCLK[3:0] bits located in Section 19.11.2, "Timer x Clock and Event Control Register," on page 317.

The Event input is synchronized to FCLK and (if enabled) filtered by a three stage filter. The resulting recreated clock is used to clock the timer in Event mode. In Bypass Mode (Sync Only), the pulse width of the external signal must be at least 2x the pulse width of the FCLK source. If the Event input not in Bypass Mode (Sync and Filter), the pulse width of the external signal must be at least 4x the pulse width of the sync and filter clock.

19.5.3 RESET

This block is reset on a nSYS_RST. On nSYS_RST all timers are reset to their default values. The timers are also reset by the RESET bit in each Timer x Control Register.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

19.6 Interrupts

The timers in the MEC1609/MEC1609i can be used to generate interrupts when the timer overflows or underflows. The timer interrupts are routed to the TIMER3, TIMER2, TIMER1, and TIMER0 bits in GIRQ15 Source Register.

Note: No interrupts are generated while the ENABLE bit is cleared.

19.7 Low Power Modes

This block is designed to conserve power when it is either sleeping or a clock source is not required.

During normal operation, if the timer is disabled via the PD bit the TIMERx_CLK_REQ signal is de-asserted. This indicates to the clock generator logic that this timer does not require the 64.52MHz clock source.

During Sleep modes the clock input is gated, the TIMERx_CLK_REQ signal is asserted, and the interrupt output goes to the inactive state. When the block returns from sleep, if enabled, it will be restarted from the preload value.

Note: The timer is terminated one TCLK after the SLEEP ENABLE is asserted.

The following table illustrates the low power mode options.

TABLE 19-3: BLOCK CLOCK GATING IN LOW POWER MODES

Power Down (PD) Bit	SLEEP ENABLE	Block Idle Status	TIMERX_CLK_REQ	State	Description
1	X	NOT IDLE	1	PREPARING to SLEEP	The core clock is still required for up to one Timer Clock period.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.
0	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is still required for up to one Timer Clock period.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

19.8 Noise Filter

The noise filter uses Filter Clock (FCLK) to filter the signal on the TINx pins. An external TINx pin must remain in the same state for three FCLK ticks before the internal state changes. The Filter Bypass bit in the Timer x Control Register is used to bypass the noise filter.

- The signal TIN may be optionally only synchronized, or synchronized and filtered depending on the filter bypass bit.
- The minimum FCLK period must be at least 2X the duration of the TIN signal so that signal can be reliably captured in the bypass mode.
- The minimum FCLK period must be at least 4X the duration of the TIN signal so that signal can be reliably captured in the non-bypass mode.
- In One-Shot mode, the TIN duration could be smaller than a TCLK period. The filtered signal is latched until the signal is seen in the TCLK domain. This also applies in the filter bypass mode.

19.9 Operating Modes

19.9.1 STARTING AND STOPPING

The MEC1609/MEC1609i timers can be started and stopped by setting and clearing the Timer Enable bit in the Timer Control Register in all modes, except one-shot.

19.9.2 TIMER MODE

The Timer mode of the MEC1609/MEC1609i is used to generate periodic interrupts to the EC. When operating in this mode the timer always counts down based on one of the internally generated clock sources. The Timer mode is selected by setting the Timer Mode Select bits in the Timer Control Register. See Section 19.11.1, "Timer x Control Register," on page 315.

The period between timer interrupts and the width of the output pulse is determined by the speed of the clock source, the clock divide ratio and the value programmed into the Timer Reload Register. The timer clock source and clock rate are selected using the Clock Source Select bits (TCLK) in the Timer x Clock and Event Control Register. See Section 19.11.2, "Timer x Clock and Event Control Register," on page 317.

TABLE 19-4: TIMER MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, “Timer Clock Frequencies,” on page 317
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, “Timer Clock Frequencies,” on page 317
Count Operation	Down Counter
Reload Operation	When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh.
Count Start Condition	UPDN = 0 (timer only mode): ENABLE = 1 UPDN = 1 (timer gate mode): ENABLE = 1 & TIN = 1;
Count Stop Condition	UPDN = 0: ENABLE = 0; UPDN = 1: (ENABLE= 0 | TIN = 0)
Interrupt Request Generation Timing	When timer underflows from 0000h to reload value (as determined by RLOAD) an interrupt is generated.
TINx Pin Function	Provides timer gate function
TOUTx Pin Function	TOUT toggles each time the timer underflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows. When the timer is running, values written to the Timer Reload Register are written to the timer counter when the timer underflows. The assertion of Reset also copies the Timer Reload Register into the timer counter.
Selectable Functions	• Reload timer on underflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Timer can be started and stopped by the TINx input pin (Gate Function) • The TOUTx pin changes polarity each time the timer underflows (Pulse Output Function)

19.9.2.1 Timer Mode Underflow

The MEC1609/MEC1609i timers operating in Timer mode can underflow in two different ways. One method, the Reload mode shown in Figure 19-2, is to reload the value programmed into the Reload register and continue counting from this value. The second method, Free Running mode Figure 19-3, is to set the timer to FFFFh and continue counting from this value. The underflow behavior is controlled by the RLOAD bit in the Timer Control Register.

FIGURE 19-2: RELOAD MODE BEHAVIOR

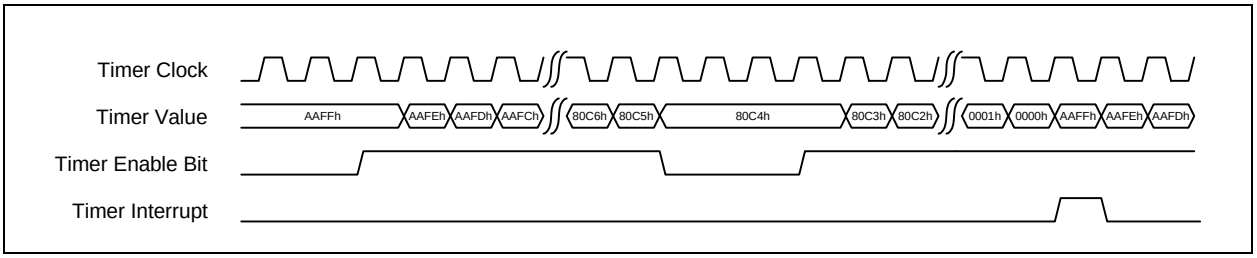
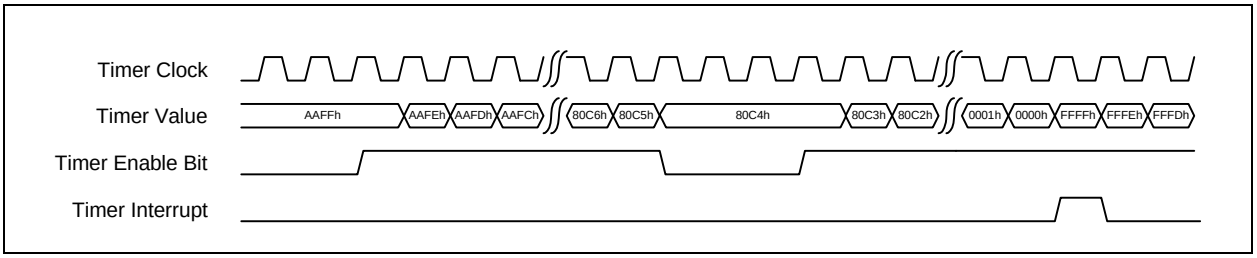


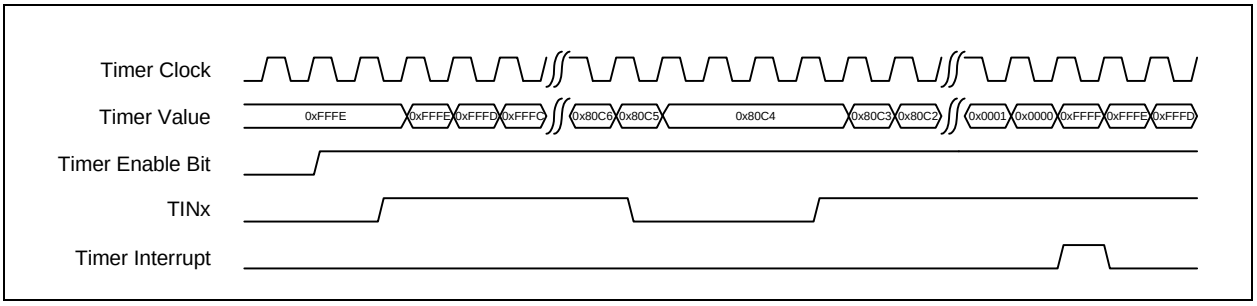
FIGURE 19-3: FREE RUNNING MODE BEHAVIOR



19.9.2.2 Timer Gate Function

The TINx pin on each timer can be used to pause the timer's operation when the timer is running. The timer will stop counting when the TINx pin is deasserted and count when the TINx pin is asserted. Figure 19-4 shows the timer behavior when the TINx pin is used to gate the timer function. The UPDN bit is used to enable and disable the Timer Gate function when in the Timer mode.

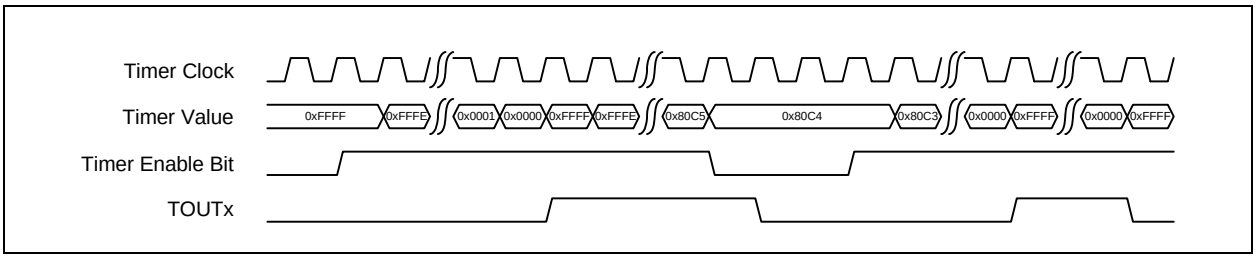
FIGURE 19-4: TIMER GATE OPERATION



19.9.2.3 Timer Mode Pulse Output

The four Timers can be used to generate a periodic output pulse. The output pulse changes state each time the timer underflows. The output is also cleared when the EN bit is cleared. Figure 19-5 shows the behavior of the TOUTx pin when it is used as a pulse output pin.

FIGURE 19-5: TIMER PULSE OUTPUT



19.9.3 EVENT MODE

Event mode is used to count events that occur external to the timer. The timer can be programmed to count the overflow output from the previous timer or an edge on the TINx pin. The direction the timer counts in Event mode is controlled by the UPDN bit in the Timer Control Register. When the timer is in Event mode, the TOUTx signal can be used to generate a periodic output pulse when the timer overflows or underflows. Figure 19-5 illustrates the pulse output behavior of the TOUTx pin in event mode when the timer underflows.

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

TABLE 19-5: EVENT MODE OPERATIONAL SUMMARY

Item	Description
Count Source	• External signal input to TINx pin (effective edge can be selected by software) • Timer x-1 overflow
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, “Timer Clock Frequencies,” on page 317
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, “Timer Clock Frequencies,” on page 317
Count Operation	Up/Down Counter
Reload Operation	• When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh. • When the timer overflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to 0000h.
Count Start Condition	Timer Enable is set (ENABLE = 1)
Count Stop Condition	Timer Enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	When timer overflows or underflows
TINx Pin Function	Event Generation
TOUTx Pin Function	TOUT toggles each time the timer underflows/overflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register

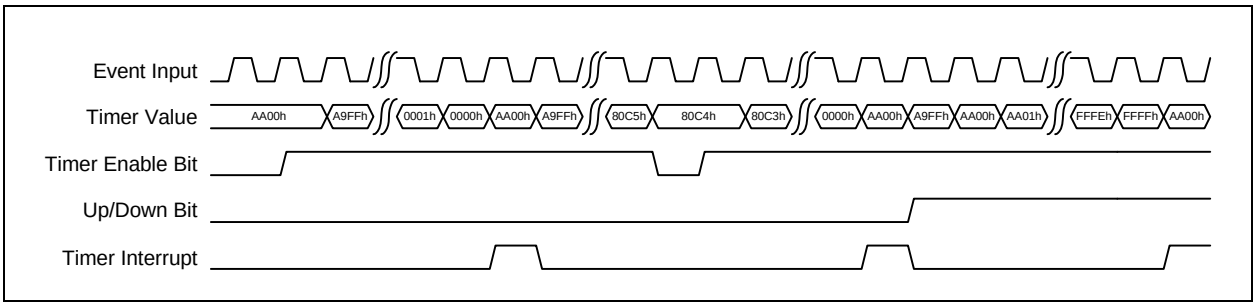
TABLE 19-5: EVENT MODE OPERATIONAL SUMMARY (CONTINUED)

Item	Description
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• The direction of the counter is selectable via the UPDN bit. • Reload timer on underflow/overflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Pulse Output Function The TOUTx pin changes polarity each time the timer underflows or overflows.

19.9.3.1 Event Mode Operation

The timer starts counting events when the ENABLE bit in the Timer Control Register is set and continues to count until the ENABLE bit is cleared. When the ENABLE bit is set, the timer continues counting from the current value in the timer except after a reset event. After a reset event, the timer always starts counting from the value programmed in the Reload Register if counting down or from 0000h if counting up. Figure 19-6 shows an example of timer operation in Event mode. The RLOAD bit controls the behavior of the timer when it underflows or overflows.

FIGURE 19-6: EVENT MODE OPERATION



19.9.4 ONE-SHOT MODE

The One-Shot mode of the timer is used to generate a single interrupt to the EC after a specified amount of time. The timer can be configured to start using the ENABLE bit (Figure 19-7) or on a timer overflow event from the previous timer. See Section 19.11.2, "Timer x Clock and Event Control Register," on page 317 for configuration details. The ENABLE bit must be set for an event to start the timer. The ENABLE bit is cleared one clock after the timer starts. The timer always starts from the value in the Reload Register and counts down in One-Shot mode.

TABLE 19-6: ONE SHOT MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, “Timer Clock Frequencies,” on page 317
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, “Timer Clock Frequencies,” on page 317
Count Operation	Down Counter
Reload Operation	When the timer underflows the timer will stop. When the timer is enabled timer starts counting from value programmed in Timer Reload Register. (RLOAD has no effect in this mode)
Count Start Condition	Setting the ENABLE bit to 1 starts One-Shot mode. The timer clock automatically clears the enable bit one timer tick later. **Note:** One-Shot mode may be enabled in Event Mode. In Event mode an overflow from the previous timer is used for timer tick rate.

TABLE 19-6: ONE SHOT MODE OPERATIONAL SUMMARY (CONTINUED)

Item	Description
Count Stop Condition	• Timer is reset (RESET = 1) • Timer underflows
Interrupt Request Generation Timing	When an underflow occurs.
TINx Pin Function	One Shot External input
TOUTx Pin Function	The TOUTx pin is asserted when the timer starts and de-asserted when the timer stops
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• Pulse Output Function The TOUTx pin is asserted when the timer starts and de-asserted when the timer stops.

FIGURE 19-7: TIMER START BASED ON ENABLE BIT

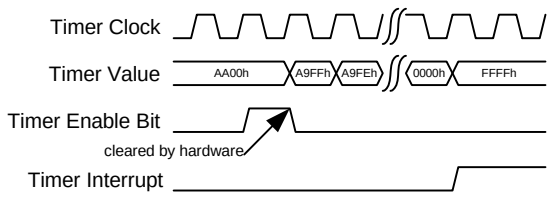


FIGURE 19-8: TIMER START BASED ON EXTERNAL EVENT

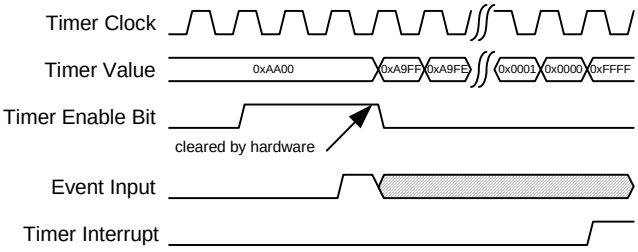
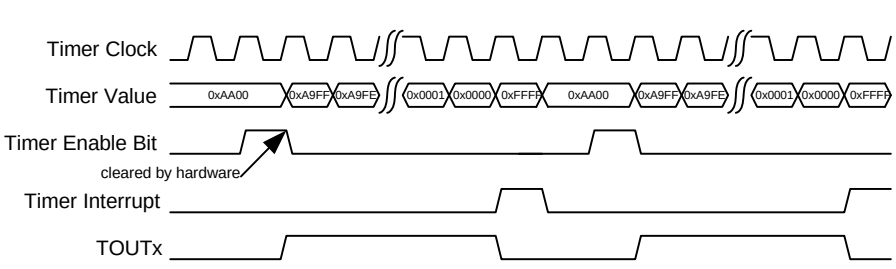


FIGURE 19-9: ONE SHOT TIMER WITH PULSE OUTPUT



19.9.5 MEASUREMENT MODE

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

TABLE 19-7: MEASUREMENT MODE OPERATIONAL SUMMARY

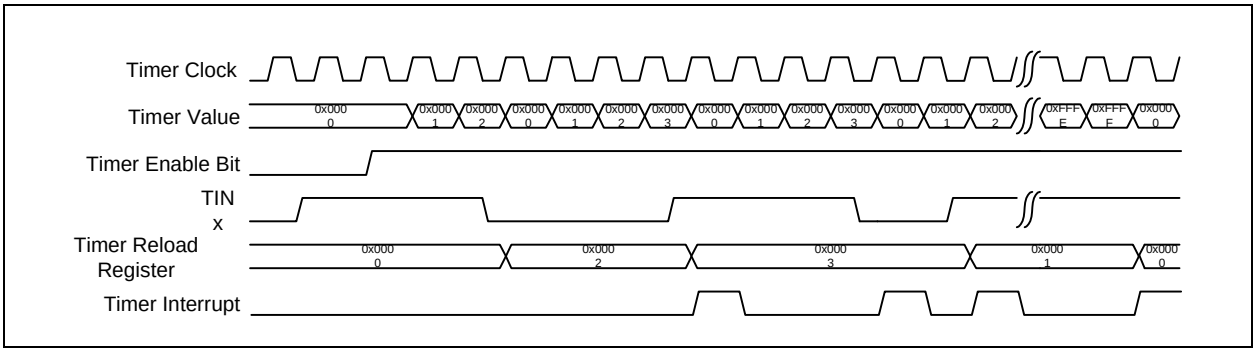
Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, "Timer Clock Frequencies," on page 317
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 19-12, "Timer Clock Frequencies," on page 317
Count Operation	• Up Count • At measurement pulse's effective edge, the count value is transferred to the Timer Reload Register and the timer is loaded with 0000h and continues counting.
Count Start Condition	• Timer enable is set (ENABLE = 1)
Count Stop Condition	• Timer is reset (RESET = 1) • Timer overflows • Timer enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	• When timer overflows • When a measurement pulse's effective edge is input. (An interrupt is not generated on the first effective edge after the timer is started.)
TINx Pin Function	Programmable Input port or Measurement input
Read From Timer	When the Timer x Reload Register is read it indicates the measurement result from the last measurement made. The Timer x Reload Register reads 0000h if the timer overflows before a measurement is made.
Write to Timer	Timer x Reload Register is Read-Only in Measurement mode

19.9.5.1 Pulse Width Measurements

The timers measure pulse width by counting the number of timer clocks since the last rising or falling edge of the TINx input. To measure the pulse width of a signal on the TINx pin, the EDGE bits in the Clock and Event Control Register, must be set to start counting on rising and falling edges. The timer starts measuring on the next edge (rising or falling) on the TINx pin after the ENABLE bit is set. The Reload register stores the result of the last measurement taken. If the timer overflows, 0x0000 is written to the Reload register and the ENABLE bit is cleared stopping the timer. Figure 19-10 shows the timer behavior when measuring pulse widths.

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

FIGURE 19-10: PULSE WIDTH MEASUREMENT

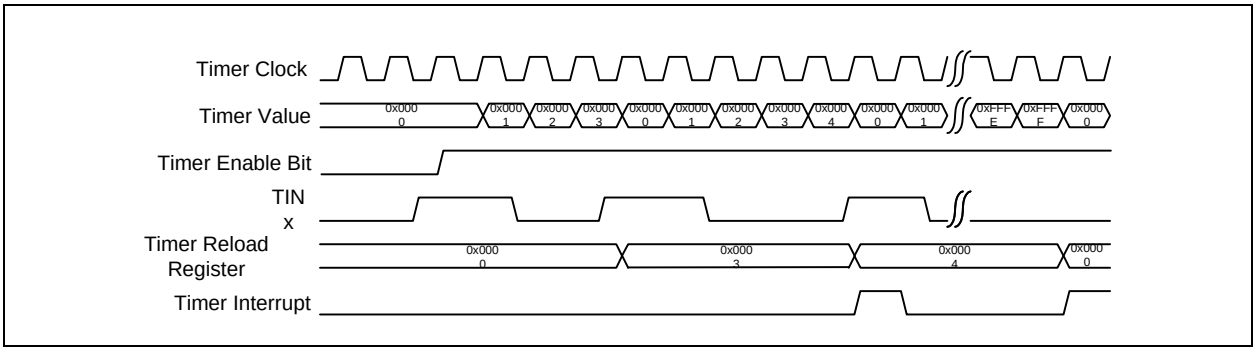


19.9.5.2 Period Measurements

The timers in the MEC1609/MEC1609i measure the period of a signal by counting the number of timer clocks between either rising or falling edges of the TINx input. The measurement edge is determined by the EDGE bits in the Clock and Event Control Register. The timer starts measuring on the next edge (rising or falling) on the TINx pin after the ENABLE bit is set. The reload register stores the result of the last measurement taken. If the timer overflows, 0x0000 is written to the reload register. Figure 19-11 shows the timer behavior when measuring the period of a signal.

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

FIGURE 19-11: PULSE PERIOD MEASUREMENT



19.10 16-Bit Counter/Timer Interface Register Summary

There are four instances of the 16-Bit Timer Interface block implemented in the MEC1609/MEC1609i enumerated as [0:3] with an overflow/underflow interface. Each instance of the 16-Bit Timer Interface has its Base Address as indicated in Table 19-8.

TABLE 19-8: 16-BIT COUNTER/TIMER INTERFACE BASE ADDRESS TABLE

16-Bit Timer Interface **Instance**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
16-bit Timer**.0**	3h	F0_0C00h
16-bit Timer**.1**		F0_0C80h = F0_0C00h + 80h
16-bit Timer**.2**		F0_0D00h = F0_0C00h + 100h
16-bit Timer**.3**		F0_0D80h = F0_0C00h + 180h

Table 19-9 is a register summary for one instance of the 16-Bit Timer Interface.

TABLE 19-9: 16-BIT COUNTER/TIMER INTERFACE REGISTER SUMMARY

	EC Interface		
Register Name	**SPB Offset**	**EC Type**	**Notes**
Timer x Control Register	00h	R/W	
Timer x Clock and Event Control Register	04h	R/W	
Timer x Reload Register	08h	R/W	
Timer x Count Register	0Ch	R	

19.11 Detailed Register Descriptions

19.11.1 TIMER X CONTROL REGISTER

TABLE 19-10: TIMER X CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			0200h			**VTR POR DEFAULT**	
	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved			TIMERx_ CLK_ REQ	SLEEP_E NABLE	TOUT Polarity	PD	Filter Bypass
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	RLOAD	TOUT_ EN	UPDN	INPOL	MODE		RESET	ENABLE

ENABLE

Timer Enable - This bit is used to start and stop the timer. This bit does not reset the timer count but does reset the timer pulse output. This bit will be cleared when the timer starts counting in One-Shot mode.

0=Timer is disabled

1=Timer is enabled

Note: The ENABLE bit is cleared after the RESET cycle is done. Firmware must poll the RESET bit.

RESET

Timer Reset - This bit stops the timer and resets the internal counter to the value in the Timer Reload Register. This bit also clears the Timer Enable bit if it is set. This bit is self clearing after the timer is reset. Firmware must poll this RESET bit.

0=Normal timer operation

1=Timer reset

APPLICATION NOTE: When the RESET takes effect interrupts are blocked. Interrupts are not blocked until RESET takes effect and the ENABLE bit is cleared. If interrupts are not desired, firmware must mask interrupt in the interrupt block.

MODE

Timer Mode Select - These bits control the timer mode.

00=Timer Mode

01=Event Mode

10=One Shot Mode

11=Measurement Mode

INPOL

Timer Input Polarity. This bit selects the polarity of the TINx input.

0=TINx input is active low (inverted)

1=TINx input is active high (non-inverted)

UPDN

Up/Down. In Event mode this bit selects the timer count direction.

Event Mode:

0=The timer counts down

1=The timer counts up

Timer Mode:

0=TINx pin has no effect on the timer

1=TINx pin pauses the timer when deasserted

TOUT_EN

TOUT Enable

0=TOUT pin is pin in the inactive state (driven low)

1=TOUT function is enabled

RLOAD

Reload Control. This bit controls how the timer is reloaded on overflow or underflow in Event and Timer modes, it has no effect in One Shot mode.

0=Roll timer over to FFFFh and continue counting when counting down and rolls over to 0000h and continues counting when counting up.

1=Reload timer from Timer Reload Register and continue counting.

FILTER BYPASS

Filter Bypass permits TINx to bypass the noise filter and go directly into the timer.

0=Filter enabled on TINx (default)

1=Filter bypassed on TINx

PD

Power Down.

0=The timer is in a running state.

1=The timer is powered down and all clocks are gated (default).

TOUT POLARITY

This bit determines the polarity of the TOUT signal. In timer modes that toggle the TOUT signal, this polarity bit will not have a perceivable difference, except to determine the inactive state. in One-Shot mode this determines if the pulsed output is active high or active low.

0=Active high (default)

1=Active low

SLEEP ENABLE

This bit is a read-only bit that reflects the state of the SLEEP ENABLE signal. This signal stops the timer and resets the internal counter to the value in the Timer Reload Register. Once the timer is disabled, the TIMERX_CLK_REQ bits will be deasserted. This signal does not clear the Timer Enable bit if it is set. If the timer is enabled, the counter will resume operation when the SLEEP ENABLE signal is deasserted. The timer is held in reset as long as the input signal is asserted.

0=Normal timer operation. In Normal Mode, the timer operates as configured. When returning from a sleep mode, if enabled, the counter will be restarted from the preload value.

1=Sleep Mode Requested. In Sleep Mode, the timer is reset, the counter is disabled, and the TIMERX_CLK_REQ outputs are deasserted.

TIMERX_CLK_REQ

The TIMERX_CLK_REQ bit is a read-only bit that reflects the state of the TIMERX_CLK_REQ output signal.

0=Indicates the 64.52MHz clock domain can be turned 'off' when appropriate

1=Indicates the 64.52MHz clock domain is required to be 'on.'

19.11.2 TIMER X CLOCK AND EVENT CONTROL REGISTER

TABLE 19-11: TIMER X CLOCK AND EVENT CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				FCLK			
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
BIT NAME	EVENT	EDGE		Reserved	TCLK			

TCLK

This field is the Timer Clock Select, used to determine the clock source to the 16-bit timer. Available frequencies are shown in Table 19-12:

TABLE 19-12: TIMER CLOCK FREQUENCIES

Timer Clock Select	Frequency Selected
0000	64.52MHz
0001	32.26MHz
0010	16.13MHz
0011	8.06MHz
0100	4.03MHz
0101	2.01MHz
0110	1MHz

TABLE 19-12: TIMER CLOCK FREQUENCIES (CONTINUED)

Timer Clock Select	Frequency Selected
0111	500KHz
1xxx	Reserved

EDGE

Edge Type Select. These bits are used to select the edge type that the timer counts. In One-Shot mode these bits select which edge starts the timer.

Event Mode:

00=Counts falling edges

01=Counts rising edges

10=Counts rising and falling edges

11=No event selected

One-Shot Mode:

00=Starts counting on a falling edge

01=Starts counting on a rising edge

10=Starts counting on a rising or falling edge

11=Start counting when the Enable bit is set

Measurement Mode:

00=Measures the time between falling edges

01=Measures the time between rising edges

10=Measures the time between rising edges and falling edges and the time between falling edges and rising edges

11=No event selected

EVENT

Event Select - This bit is used to select the count source when the timer is operating in event mode.

0=Timer x-1 overflow is count source

1=TINx is count source

FCLK

This field is the Filter Clock Select, used to determine the clock source for the TINx noise filter. Available frequencies are the same as the Timer clock and are shown in Table 19-12.

19.11.3 TIMER X RELOAD REGISTER

TABLE 19-13: TIMER X RELOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			**VTR POR DEFAULT**	
	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Timer Reload [15:0]							

TIMER RELOAD

The Timer Reload register is used in Timer and One-Shot modes to set the lower limit of the timer. In Event mode the Timer Reload register sets either the upper or lower limit of the timer depending on if the timer is counting up or down. Valid Timer Reload values are 0001h - FFFFh. If the timer is running, the reload value will not be updated until the timer overflows or underflows.

Note:	Programming a 0000h as a preload value is not a valid count value.

19.11.4 TIMER X COUNT REGISTER

TABLE 19-14: TIMER X COUNT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	0Ch			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			**VTR POR DEFAULT**	
	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Timer Count [15:0]							

TIMER COUNT

The Timer Count register returns the current value of the timer in all modes.

20.0 HIBERNATION TIMER

20.1 General Description

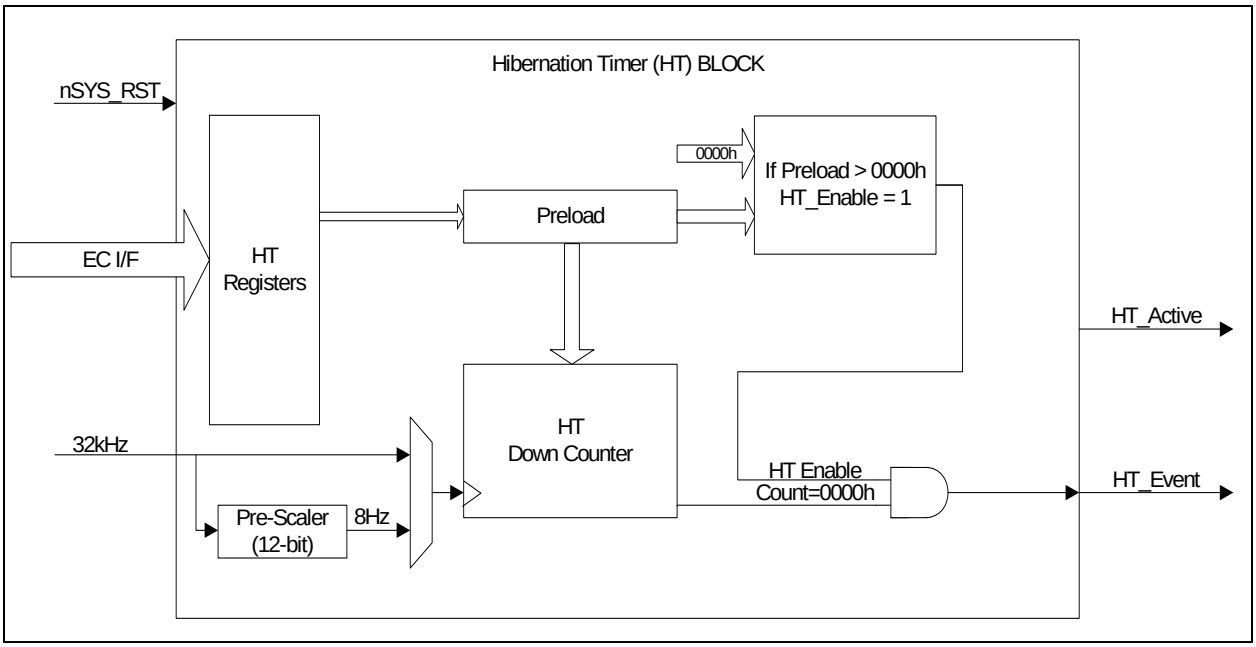
The Hibernation Timer can generate a wake event to the Embedded Controller (EC) when it is in a hibernation mode. This block supports wake events up to 2 hours in duration. The timer is a 16-bit binary count-down timer that can be programmed in 30.5us and 0.125 second increments for period ranges of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

See GIRQ23 Source Register on page 287 for details on enabling the Hibernation Timer wake-up event.

It takes up to 1 32-kHz clock period for registers to get updated after register writes. The ring oscillator must not be stopped for at least one 32-kHz clock following a register write.

20.2 Block Diagram

FIGURE 20-1: HIBERNATION TIMER BLOCK DIAGRAM



20.3 Block Diagram Signal List

TABLE 20-1: BLOCK DIAGRAM SIGNAL LIST DESCRIPTION

Signal Name	Direction	Description
nSYS_RST	INPUT	VTR Power on Reset.
X32K_CLK	INPUT	32Khz, Clock Source for Hibernation Timer
HT_Active	OUTPUT	Signal indicating that the timer is enabled and actively counting
HT_Event	OUTPUT	Signal indicating that the timer is enabled and has expired. This signal is used to generate an Hibernation Timer interrupt event.
E/C IF	I/O Bus	Bus used by microprocessor to access the registers in this block.

20.4 Power, Clocks and Reset

20.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

20.4.2 CLOCKS

This block has two clock inputs, the EC Bus Clock, and X32K_CLK. The EC Bus Clock is used in the interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK is the clock source for the Hibernation Timer functional logic, including the counters.

MCLKX32K_CLKSee Section 5.4, "Clock Generator," on page 76 for details on clocks.

20.4.3 RESET

This block is reset on a nSYS_RST.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

20.5 Interrupts

Each instance of the Hibernation Timer in the MEC1609/MEC1609i can be used to generate interrupts and wake-up events when the timer underflows. The Hibernation Timer interrupts are routed to the HTIMER1 & HTIMER0 bits in GIRQ23 Source Register on page 287.

20.6 Registers

There are two instances of Hibernation Timer block implemented in the MEC1609/MEC1609i enumerated as.Hibernation Timer.0 & Hibernation Timer.1. Each instance of the Hibernation Timer has its own Logical Device Number, and Base Address as indicated in Table 20-2.

TABLE 20-2: Hibernation Timer BASE ADDRESS TABLE

Hibernation Timer **Instance**	**LDN from (**Table 3-2 on page 48**)**	**AHB Base Address**
Hibernation Timer**.0**	0h	F0_0000h
Hibernation Timer**.1**		F0_0000h + 80h

The Table 20-3 is a register summary for one instance of the Hibernation Timer. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 20-3: Hibernation Timer REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
HTimer x Preload Register	00h	1-0	R/W	
Hibernation Timer x Control Register	04h	0	R/W	
Hibernation Timer x Count Register	08h	1-0	R	

20.6.1 HTIMER X PRELOAD REGISTER

TABLE 20-4: HTIMER X PRELOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	HT Preload[15:0]							

HT PRELOAD[15:0]

This register is used to set the Hibernation Timer Preload value. Writing this register to a non-zero value resets the down counter to start counting down from this programmed value. Writing this register to 0000h disables the hibernation counter. The resolution of this timer is determined by the CTRL bit in the HTimer x Control Register.

Since the timer runs off of the X32K_CLK clock, up to two 32-kHz clock periods are needed for the new load value to be stored into the counter. This is especially important when the system is put to sleep after software reloads the counter: the system clock might be stopped before the counter is updated with new value, hence keeping the counter in reload mode and disabling counting. It is recommended that software to poll the Hibernation Timer x Count Register until its value reflects that of the new load before entering sleep mode.

20.6.2 HTIMER X CONTROL REGISTER

TABLE 20-5: HIBERNATION TIMER X CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							R/W
BIT NAME	Reserved							CTRL

CTRL

0= The Hibernation Timer has a resolution of 30.5us per LSB, which yields a maximum time of ~2seconds.

1= The Hibernation Timer has a resolution of 0.125s per LSB, which yields a maximum time in excess of 2 hours.

20.6.3 HTIMER X COUNT REGISTER

TABLE 20-6: HIBERNATION TIMER X COUNT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							
BIT NAME	Count[15:0]							

COUNT[15:0]

The current state of the Hibernation Timer.

21.0 WEEK ALARM INTERFACE

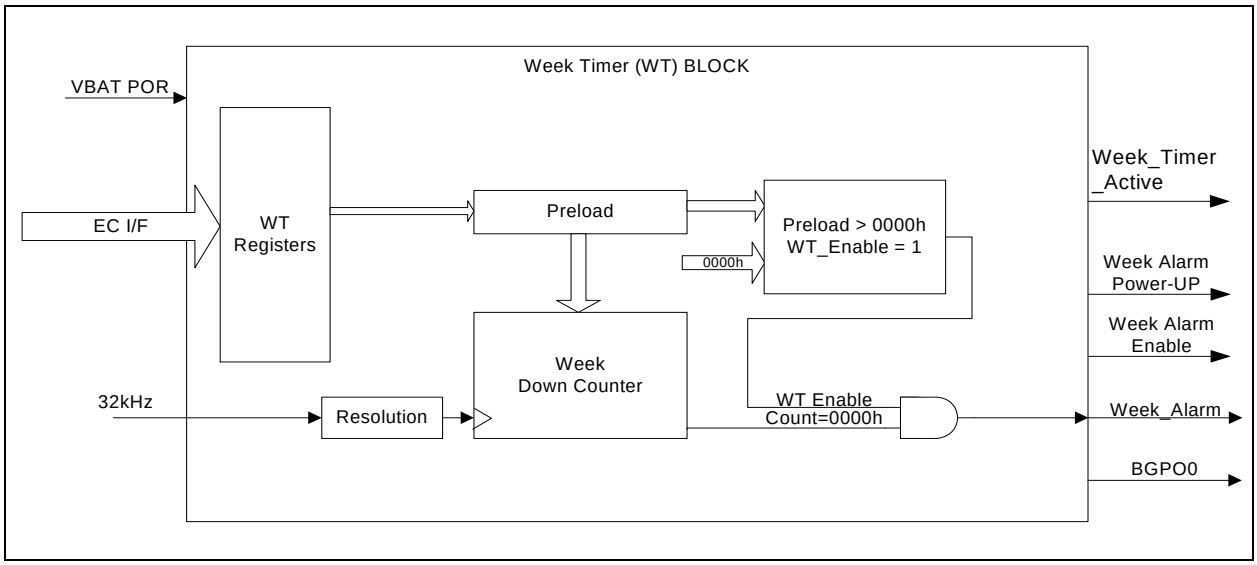
21.1 General Description

The Week Alarm Interface provides a 16-bit, battery-powered, Week Timer that supports 1 ms, 1 second and 1 minute resolution and auto reloads following a counter underflow (Figure 21-1). In addition, this block interfaces directly with the VBAT-Powered Control Interface and includes a VBAT-backed general-purpose output pin (BGPO0).

It takes up to two X32K_CLK clock period for registers to get updated after register writes. The 64.52 MHz Ring Oscillator must not be stopped for at least one X32K_CLK clock period following a register write.

21.2 Block Diagram

FIGURE 21-1: WEEK TIMER BLOCK DIAGRAM



21.3 Signal List for Block Diagram

TABLE 21-1: WEEK ALARM INTERFACE SIGNAL LIST

Signal Name	Direction	Description
VBAT_POR	Input	VTR Power on Reset.
EC Bus Clock	Input	Bus Clock (part of the EC Interface)
X32K_CLK	Input	Core logic clock
Week Alarm Power-Up Output	Output	Week Timer wake up event signal
Week_alrm_en	Output	Output to control the function of the Week Timer Output
Week_alarm	Output	Week Timer Interrupt indicating that the timer has expired.
Week_Timer_Active	Output	The Week_Timer_Active output is asserted when the counter is enabled and counting. It is cleared when the Week_Timer is disabled or not counting.
EC Interface	I/O Bus	Bus used by microprocessor to access the registers in this block.
BGPO0	Output	VBAT-powered General Purpose Output (see the BGPO0 bit D5 in the Week Timer Control Register).

21.4 Power, Clocks and Reset

21.4.1 POWER DOMAIN

This block is powered by the VBAT power supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

21.4.2 CLOCKS

This block has two clock inputs: EC Bus Clock and the X32K_CLK. EC Bus Clock is used by the EC Data Memory Bus to interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK clock source is the clock source for the week alarm logic.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

21.4.3 POWER ON RESET

This block is reset on VBAT_POR.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

21.5 Interrupts

The Week Alarm Interface generates an interrupt and wakeup event following a Week Timer underflow. The Interrupt is routed WEEK_ALR in GIRQ23 Source Register. The WEEK_ALR wake and interrupt event can be asserted when VBAT is powered and VTR is unpowered. The WEEK_ALR wake event and Interrupt event is retained during VBAT and is detected after the next VTR POR power sequence.

21.6 Week Timer

The Week_Timer_Active output is asserted when the counter is enabled and counting. It is cleared when the Week_-Timer is disabled or not counting.

21.7 Week Alarm Power-Up Output

The internal Week Alarm Power-Up Output signal drives an input to the VBAT-Powered Control Interface as described in Section 32.0, "VBAT-Powered Control Interface," on page 437. The Week Alarm Power-Up Output signal is driven even when the VTR supply is unpowered.

The WEEK_ALRM_EN bit in the Week Timer Control Register enables the Week Alarm Power-Up Output function in the VBAT-Powered Control Interface (see FIGURE 32-1: on page 437). Once the internal Week Alarm Power-Up Output signal drives the input to the VBAT-Powered Control Interface as a result of an Interrupts, the WEEK_ALRM_EN bit must be cleared to reset the Week Alarm Interface.

APPLICATION NOTE: The WEEK_ALRM_EN bit defaults to ‘0,’ disabling the ability to power the system up by the Week Alarm Interface. This is necessary to avoid an uninitialized Week Alarm Interface from causing unintended power-up events.

21.8 Registers

The Week Alarm Interface has its own Logical Device Number, and Base Address as indicated in Table 21-2. Table 21-3 is a register summary for the Week Alarm Interface block. See Note 3-1 on page 49.

TABLE 21-2: Week Alarm Interface BASE ADDRESS TABLE

Week Alarm Interface Blocks	LDN from (Table 3-2 on page 48)	AHB Base Address
Week Alarm Timer	33h	F0_CC80h

Table 21-3 is a register summary for the Week Alarm Interface block. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 21-3: Week Alarm Interface REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Week Timer Control Register	00h	0	R/W	
Week Timer Reload Register	04h	0:1	R/W	
Week Timer Data Register	08h	0:1	R	

21.8.1 WEEK TIMER CONTROL REGISTER

The Week Timer Control Register is used to configure the Week Timer.

TABLE 21-4: WEEK TIMER CONTROL REGISTER

HOST ADDRESS	n/a					n/a		**HOST SIZE**
EC OFFSET	00h					8-bit		**EC SIZE**
POWER	VBAT					01h		VBAT_POR **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R	R	R/W (Note 21-1)		R/W
BIT NAME	Reserved	WEEK_ALRM_EN	BGPO0	Reserved		RESOLUTION		WT_ENABLE

WT_ENABLE

Week Timer Enable - This bit is used to start and stop the Week Timer. The Week timer is held when the timer is disabled and starts counting from the value in the Week Timer Reload register when enabled.

0 - Week Timer is disabled.

1 - Week Timer is enabled.

(See Note 21-1 on page 327.)

RESOLUTION

Week Timer Resolution - These bits are used to control the resolution of the Week Timer counter.

00 - 1 minute resolution.

01 - 1 Second resolution

10 - 1 Millisecond resolution.

BGPO0

VBAT-powered General Purpose Output Control that is used as part of the VBAT-Powered Control Interface.

0= output low (default)

1= output high

WEEK_ALRM_EN

VCI Week Alarm Enable- This bit controls the routing of the Week Alarm Power-Up Output output to VBAT-Powered Control Interface to assert the VCI signal. Once the internal Week_alrm_en signal drives the input to the VBAT-Powered Control Interface as a result of Interrupts, this bit must be cleared to reset the Week Alarm Interface.

0 - Disable routing (Default)

1 - Enable routing

Note:	the Week Timer Enable bit [D0] must be cleared (‘0’) when changing the Week Timer Resolution bits. For example to change the resolution of the Week Timer two writes to the Week Timer Control Register are required: the first write de-asserts the WT_ENABLE bit, the second write modifies the RESOLUTION bits and asserts the WT_ENABLE bit.

21.8.2 WEEK TIMER RELOAD REGISTER

TABLE 21-5: WEEK TIMER RELOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VBAT			2760h			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W Note 21-1							
BIT NAME	WEEK TIMER RELOAD[15:0]							

WEEK TIMER RELOIAD[15:0]

This register contains the value that is used to reload the Week Timer Data Register when the latter underflows. A Reload value of 0000h is equivalent to a reload value of FFFFh. In both cases the Week Timer will count 2^{16} times before triggering an interrupt.

Note 21-1 The EC must clear the Enable bit in the Week Timer Control Register to perform a write access to the Week Timer Reload Register.

Note 21-2 A write to the Week Timer Reload Register of 0000h will be treated as a full count (FFFFh +1) and start downcounting when the WT_ENABLE bit in the Week Timer Control Register is set to ‘1’.

21.8.3 WEEK TIMER DATA REGISTER

TABLE 21-6: WEEK TIMER DATA REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			16-bit			**EC SIZE**	
POWER	VBAT			2760h			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							
BIT NAME	WEEK TIMER Counter[15:0]							

WEEK TIMER COUNTER[15:0]

The current state of the Week Timer.

22.0 GPIO INTERFACE

22.1 General Description

The MEC1609/MEC1609i GPIO Interface provides general purpose input monitoring and output control, as well as managing many aspects of pin functionality; including, multi-function Pin Multiplexing Control, Output Buffer Type control, PU/PD resistors, asynchronous wakeup and synchronous Interrupt Detection, GPIO Direction, and Polarity control.

Features of the GPIO Interface include:

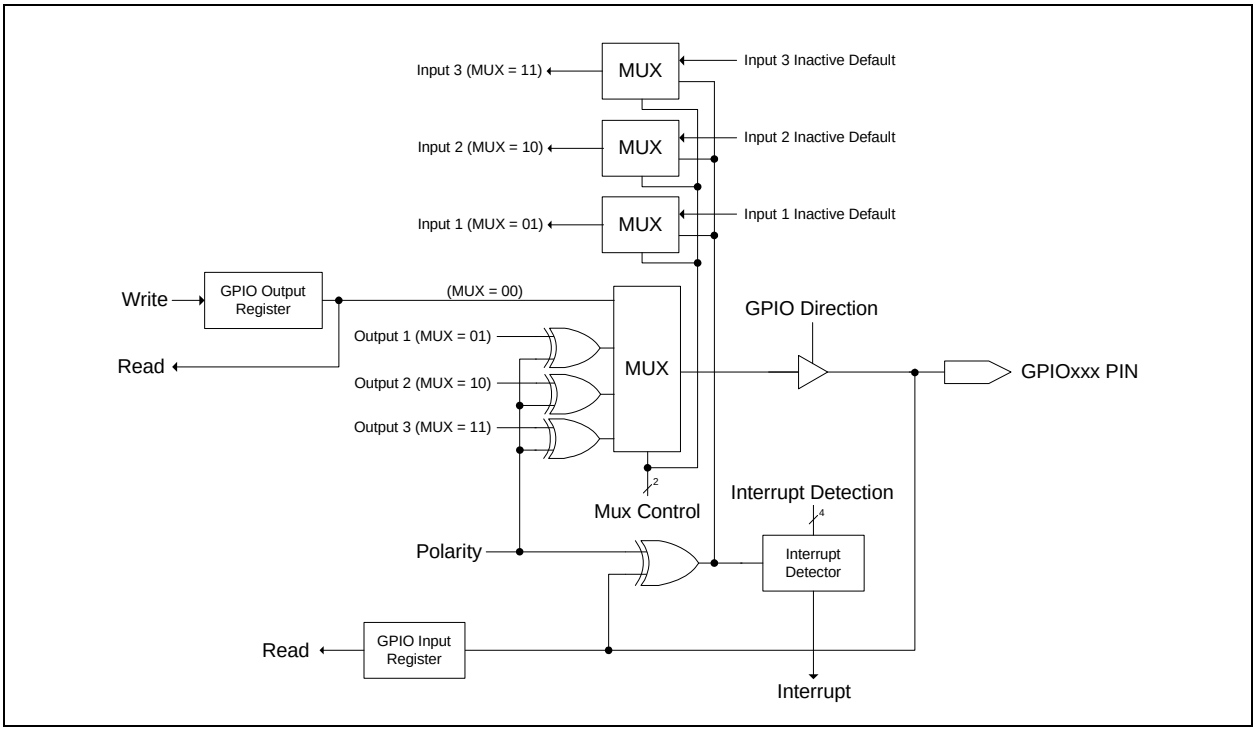
- Inputs:
  Asynchronous rising and falling edge wakeup detection
  Interrupt High or Low Level
- Outputs:
  Push Pull or Open Drain output
- Pull up or pull down resistor control
- Interrupt and wake capability available for all GPIOs
- 8 GPIO Pass-Through Ports
- Group- or individual control of GPIO data. See Section 22.3 and Section 22.11
- Multi-function Pin Multiplexing is controlled by the GPIO Interface

22.2 Block Diagram

The GPIO Interface Block Diagram shown in Figure 22-1 illustrates the functionality of a single MEC1609/MEC1609i GPIO Interface pin. The source for the Pin Multiplexing Control, Interrupt Detection, GPIO Direction, and Polarity controls in Figure 22-1 is a Pin Control Register that is associated with each pin (see Section 22.10.1, "Pin Control Register," on page 337).

The MEC1609/MEC1609i supports up to four independent signal functions per pin including the GPIO signal function itself, which is always positioned at Mux Control = '00.' The GPIO Input Registers and the GPIO Output Registers provide the GPIO Interface 'Read' and 'Write' functionality illustrated in Figure 22-1 (see Section 22.10.3, "GPIO Input Registers," on page 342 and Section 22.10.2, "GPIO Output Registers," on page 340).

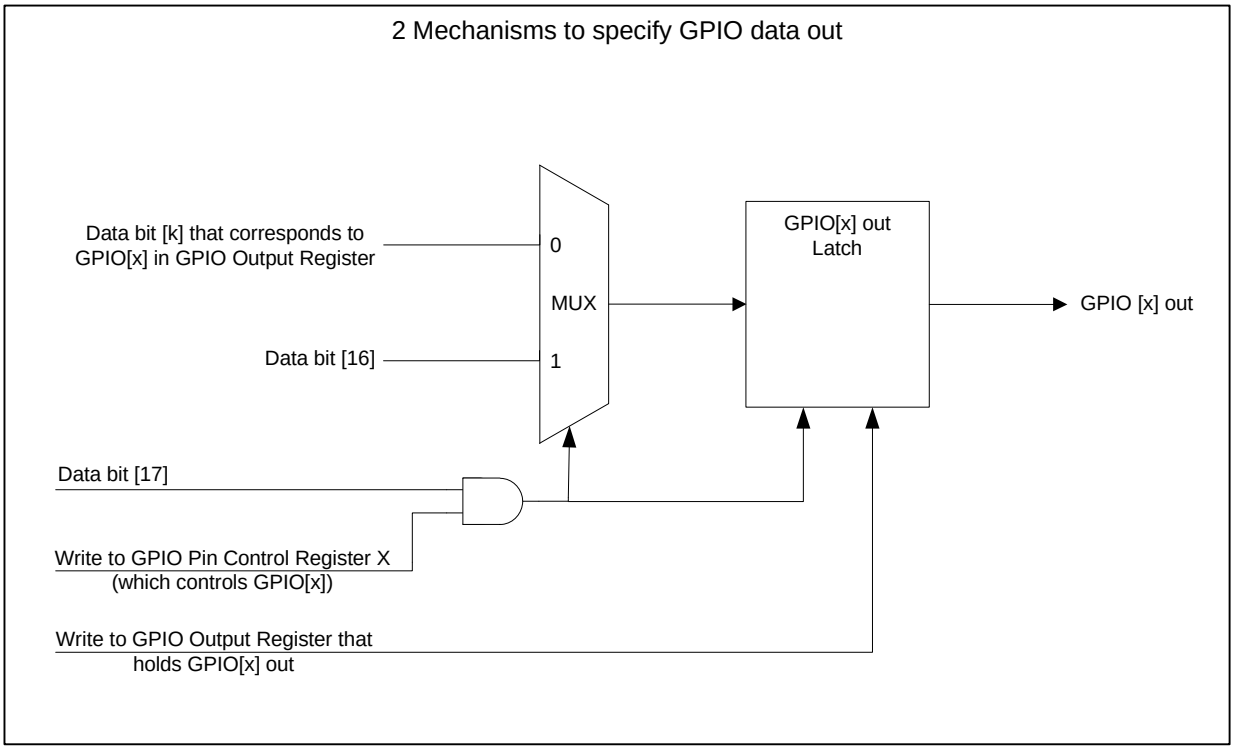
FIGURE 22-1: GPIO Interface BLOCK DIAGRAM



22.3 Accessing GPIOs

There are two ways to specify GPIO input and output port data. In the legacy approach that maintains compatibility with earlier generation devices, outputs to individual GPIO ports are grouped into four 32-bit GPIO Output Registers (see Table 22-8.) It is incumbent on firmware to modify particular bit(s) while not disturbing the others. The MEC1609/MEC1609i supports an alternative approach in which each port's output is individually specified, i.e., Bit [16] in the port Pin Control Register is used for output data. Bit [17] is used to enable this alternative write to the GPIO on a per-bit basis. Figure 22-2 illustrates the concept. On reads, Bit [16] returns the programmed value while Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [17].

FIGURE 22-2: OUTPUT DATA TO GPIO



22.4 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 MEC1609/MEC1609i GPIO Indexing is done sequentially starting from 'GPIO000.' There is a unique index number for each of the bits in the GPIO Input Registers (Table 22-1). Index numbers for the NORM Exception Bits are skipped; e.g., there is no GPIO037, GPIO077, GPIO137 or GPIO177. (See Section 22.10.3, "GPIO Input Registers," on page 342.)

TABLE 22-1: GPIO Indexing

Bit Position	GPIO[000:036]	GPIO[040:076]	GPIO[100:136]	GPIO[140:176]	GPIO[200:236]
Bit 0	GPIO000	GPIO040	GPIO100	GPIO140	GPIO200
Bit 1	GPIO001	GPIO041	GPIO101	GPIO141	GPIO201
Bit 2	GPIO002	GPIO042	GPIO102	GPIO142	GPIO202
Bit 3	GPIO003	GPIO043	GPIO103	GPIO143	GPIO203
Bit 4	GPIO004	GPIO044	GPIO104	GPIO144	GPIO204
Bit 5	GPIO005	GPIO045	GPIO105	GPIO145	GPIO205

TABLE 22-1: GPIO Indexing (CONTINUED)

Bit Position	GPIO[000:036]	GPIO[040:076]	GPIO[100:136]	GPIO[140:176]	GPIO[200:236]
Bit 6	GPIO006	GPIO046	GPIO106	GPIO146	GPIO206
Bit 7	GPIO007	GPIO047	GPIO107	GPIO147	GPIO207
Bit 8	GPIO010	GPIO050	GPIO110	GPIO150	GPIO210
Bit 9	GPIO011	GPIO051	GPIO111	GPIO151	GPIO211
Bit 10	GPIO012	GPIO052	GPIO112	GPIO152	GPIO212
Bit 11	GPIO013	GPIO053	GPIO113	GPIO153	GPIO213
Bit 12	GPIO014	GPIO054	GPIO114	GPIO154	GPIO214
Bit 13	GPIO015	GPIO055	GPIO115	GPIO155	GPIO215
Bit 14	GPIO016	GPIO056	GPIO116	GPIO156	GPIO216
Bit 15	GPIO017	GPIO057	GPIO117	GPIO157	GPIO217
Bit 16	GPIO020	GPIO060	GPIO120	GPIO160	GPIO220
Bit 17	GPIO021	GPIO061	GPIO121	GPIO161	GPIO221
Bit 18	GPIO022	GPIO062	GPIO122	GPIO162	GPIO222
Bit 19	GPIO023	GPIO063	GPIO123	GPIO163	GPIO223
Bit 20	GPIO024	GPIO064	GPIO124	GPIO164	GPIO224
Bit 21	GPIO025	GPIO065	GPIO125	GPIO165	GPIO225
Bit 22	GPIO026	GPIO066	GPIO126	GPIO166	GPIO226
Bit 23	GPIO027	GPIO067	GPIO127	GPIO167	GPIO227
Bit 24	GPIO030	GPIO070	GPIO130	GPIO170	GPIO230
Bit 25	GPIO031	GPIO071	GPIO131	GPIO171	GPIO231
Bit 26	GPIO032	GPIO072	GPIO132	GPIO172	GPIO232
Bit 27	GPIO033	GPIO073	GPIO133	GPIO173	GPIO233
Bit 28	GPIO034	GPIO074	GPIO134	GPIO174	GPIO234
Bit 29	GPIO035	GPIO075	GPIO135	GPIO175	GPIO235
Bit 30	GPIO036	GPIO076	GPIO136	GPIO176	GPIO236
Bit 31	Reserved or Norm Exception Bit	Reserved or Norm Exception Bit	Reserved or Norm Exception Bit	Reserved or Norm Exception Bit	Reserved or Norm Exception Bit

22.5 Pin Multiplexing Control

As described above in Section 22.2, "Block Diagram", pin multiplexing depends upon the Mux Control bits in the Pin Control Register. There is a Pin Control Register for each GPIO signal function.

The MEC1609/MEC1609i Pin Control Register address offsets are shown in Table 22-2 – Table 22-6 and depend on the GPIO Index number. Pin Control Register defaults are also shown in Table 22-2 – Table 22-5. Notes for these tables are defined below.

GPIO signal function names in parentheses in Table 22-2 – Table 22-5 represent interrupt/wake-only GPIO signal functions. The Mux Control bits in the Pin Control Register for these pins should not be programmed ‘00.’

22.6 Notes for the Following Tables

Note 12	This pin has EC wakeup and interrupt capability controlled by the corresponding Pin Control Register. A GPIO assignment is documented in the GPIO chapter to provide interrupt and wakeup capability. The GPIO should not be used for I/O. See Detailed Pin Multiplexing Assignments section in the GPIO chapter and lookup this pin and see the associated note.
Note 13	The two pin debug port UART can be used by the Host or EC. This pin can be VCC protected or not VCC portected under program control by the POWER bit in the Configuration Select Register in Host configuration space (also accessible by the EC).
Note 14	When the JTAG_RST# pin is not asserted (logic'1'), the JTAG_TDI, JTAG_TDO, JTAG_CLK, 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_CLK, 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 route the JTAG interface to the pins. System Board Designer should terminate this pin in all functional state using jumpers and pull-up or pull down resistors, etc .
Note 15	All VBAT powered pins with GPIO's on then have only one direction selected by the default signal function. The associated GPIO input register, output register bits are not connected to the pin. Only the Interrupt Detection field in the associated pin control register function; the remainder of the bits in the pin control register has no effect.
Note 16	PS/2 ports ending with signal functions ending with "A" or "B" are muxed to a single controller. Only one set of clock and data are intended to used at a time (either "A" or "B" not both. The unused port segment should have its associated pin control register's, Mux Control Field programmed away from the PS2 controller.
Note 17	Most GPIO's are (I/O/OD). See Multiplexing tables below and associated notes for specific exceptions.
Note 18	The GPIO assignment on this pin only provides interrupt and wakeup capability. This is provided by the Interrupt Detection field in the Pin Control register. The Mux control field in the Pin Control Register should **not** be set to '00' = GPIO or undesirable results may occur.
Note 19	The PECI REQUEST# signal function must be configured as open-drain driver with an external pullup to VCC.
Note 20	This pin is also used as a JTAG TAP controller select strap option. There is a weak pullup enabled on this pin by default.

TABLE 22-2: PIN CONTROL MUXING PIN REFERENCE NUMBERS [1:32]

Pin Ref. Number	Ball	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11	Notes
1	D4	None		N/A	XTAL1	XTAL1	Reserved	Reserved	Reserved	
2	E3			N/A	AGND					
3	D3	None		N/A	XTAL2	XTAL2	Reserved	Reserved	Reserved	
4	C4			N/A	VBAT					
5	C1	None		N/A	BGPO0	BGPO0	Reserved	Reserved	Reserved	
6	D1	None		N/A	VCI_OUT	VCI_OUT	Reserved	Reserved	Reserved	
7	C2	(GPIO161)	01C4	00001000	VCI_IN2#	Reserved	VCI_IN2#	Reserved	Reserved	**Note 18**
8	F4	(GPIO162)	01C8	00001000	VCI_IN1#	Reserved	VCI_IN1#	Reserved	Reserved	**Note 18**
9	D2	(GPIO163)	01CC	00001000	VCI_IN0#	Reserved	VCI_IN0#	Reserved	Reserved	**Note 18**
10	E2	(GPIO164)	01D0	00001000	VCI_OVRD_IN	Reserved	VCI_OVRD_IN	Reserved	Reserved	**Note 18**
11	E1	(GPIO000)	0000	00001000	VCI_IN3#	Reserved	VCI_IN3#	Reserved	Reserved	**Note 18**
12	E7	GPIO160	01C0	00000000	GPIO160	GPIO160	32KHZ_OUT	KSO17	Reserved	
13	F3	(GPIO057)	00BC	00001000	VCC_PWRGD	Reserved	VCC_PWRGD	Reserved	Reserved	**Note 18**
14	F2	GPIO106	0118	00000000	GPIO106	GPIO106	nRESET_OUT	Reserved	Reserved	
15	F7	GPIO101	0104	00000000	GPIO101	GPIO101	ECGP_SCLK	Reserved	Reserved	
16	G7	GPIO102	0108	00000000	GPIO102	GPIO102	ECGP_SOUT	Reserved	Reserved	
17	G6	GPIO103	010C	00000000	GPIO103	GPIO103	ECGP_SIN	Reserved	Reserved	
18	G3			N/A	VSS_RO					
19	F1			N/A	VTR					
20	H6			N/A	VSS					
21	E8	GPIO021	0044	00000000	GPIO021	GPIO021	RC_ID	KSI2	Reserved	
22	G2			N/A	VTR_REG					
23	G1			N/A	VR_CAP					
24	G5	GPIO060	00C0	00000000	GPIO060	GPIO060	KBRST	Reserved	Reserved	
25	G4	GPIO127	015C	00000000	GPIO127	GPIO127	A20M	Reserved	Reserved	
26	H4	GPIO116	0138	00000001	GPIO116	GPIO116	MSDATA	Reserved	Reserved	**Note 20**
27	H5	GPIO117	013C	00000000	GPIO117	GPIO117	MSCLK	Reserved	Reserved	
28	H1			N/A	AVTR_ADC					
29	H3			N/A	VREF_ADC					
30	J3	GPIO200	0200	00001000	ADC0	GPIO200	ADC0	Reserved	Reserved	
31	H2	GPIO210	0220	00001000	ADC8	GPIO210	ADC8	Reserved	Reserved	
32	J1	GPIO201	0204	00001000	ADC1	GPIO201	ADC1	Reserved	Reserved	

TABLE 22-3: PIN CONTROL MUXING PIN REFERENCE NUMBERS [33:64]

Pin Ref. Number	Ball	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11	Notes
33	J2	GPIO211	0224	00001000	ADC9	GPIO211	ADC9	Reserved	Reserved	
34	K2	GPIO202	0208	00001000	ADC2	GPIO202	ADC2	Reserved	Reserved	
35	K1	GPIO212	0228	00001000	ADC10	GPIO212	ADC10	Reserved	Reserved	
36	L1	GPIO203	020C	00001000	ADC3	GPIO203	ADC3	Reserved	Reserved	
37	K3	GPIO213	022C	00001000	ADC11	GPIO213	ADC11	Reserved	Reserved	
38	J4			N/A	AVTR_ADC					
39	K4	GPIO204	0210	00001000	ADC4	GPIO204	ADC4	Reserved	Reserved	
40	L2	GPIO214	0230	00001000	ADC12	GPIO214	ADC12	Reserved	Reserved	
41	M1	GPIO205	0214	00001000	ADC5	GPIO205	ADC5	Reserved	Reserved	
42	M2	GPIO215	0234	00001000	ADC13	GPIO215	ADC13	Reserved	Reserved	
43	L3	GPIO206	0218	00001000	ADC6	GPIO206	ADC6	Reserved	Reserved	
44	L4	GPIO216	0238	00001000	ADC14	GPIO216	ADC14	Reserved	Reserved	
45	M3	GPIO207	021C	00001000	ADC7	GPIO207	ADC7	Reserved	Reserved	
46	M4	GPIO217	023C	00001000	ADC15	GPIO217	ADC15	Reserved	Reserved	
47	J5			N/A	VSS_ADC					
48	L5	(GPIO064)	00D0	00001000	LRESET#	Reserved	LRESET#	Reserved	Reserved	**Note 18**
49	K5	(GPIO067)	00DC	00001000	CLKRUN#	Reserved	CLKRUN#	Reserved	Reserved	**Note 18**
50	K6	(GPIO066)	00D8	00001000	LFRAME#	Reserved	LFRAME#	Reserved	Reserved	**Note 18**
51	J6	(GPIO062)	00C8	00001000	LDRQ#	Reserved	LDRQ#	Reserved	Reserved	**Note 18**
52	M5	(GPIO063)	00CC	00001000	SER_IRQ	Reserved	SER_IRQ	Reserved	Reserved	**Note 18**
53	H7			N/A	VTR					
54	M6	(GPIO065)	00D4	00001000	PCI_CLK	Reserved	PCI_CLK	Reserved	Reserved	**Note 18**
55	L6	None		N/A	LAD0	LAD0	Reserved	Reserved	Reserved	
56	L7	None		N/A	LAD1	LAD1	Reserved	Reserved	Reserved	
57	K7	None		N/A	LAD2	LAD2	Reserved	Reserved	Reserved	
58	J7	None		N/A	LAD3	LAD3	Reserved	Reserved	Reserved	
59	J8	GPIO100	0100	00000000	GPIO100	GPIO100	nEC_SCI	Reserved	Reserved	
60	M7	GPIO011	0024	00000000	GPIO011	GPIO011	nSMI	Reserved	Reserved	
61	K8	GPIO061	00C4	00000000	GPIO061	GPIO061	LPCPD#	Reserved	Reserved	
62	M8	None		N/A	nFWP	nFWP	Reserved	Reserved	Reserved	
63	L8	GPIO050	00A0	00000000	GPIO050	GPIO050	FAN_TACH0	Reserved	Reserved	
64	M9	GPIO051	00A4	00000000	GPIO051	GPIO051	FAN_TACH1	Reserved	Reserved	

TABLE 22-4: PIN CONTROL MUXING PIN REFERENCE NUMBERS [65:96]

Pin Ref. Number	Ball	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11	Notes
65	L9	GPIO052	00A8	00000000	GPIO052	GPIO052	FAN_TACH2	Reserved	Reserved	
66	K9	GPIO016	0038	00000000	GPIO016	GPIO016	GPTP-IN7	FAN_TACH3	Reserved	
67	L10	GPIO053	00AC	00000000	GPIO053	GPIO053	PWM0	Reserved	Reserved	
68	M10	GPIO054	00B0	00000000	GPIO054	GPIO054	PWM1	Reserved	Reserved	
69	M11	GPIO055	00B4	00000000	GPIO055	GPIO055	PWM2	Reserved	Reserved	
70	L11	GPIO056	00B8	00000000	GPIO056	GPIO056	PWM3	Reserved	Reserved	
71	M12	GPIO001	0004	00000000	GPIO001	GPIO001	PWM4	Reserved	Reserved	
72	K10	GPIO002	0008	00000000	GPIO002	GPIO002	PWM5	Reserved	Reserved	
73	L12	GPIO014	0030	00000000	GPIO014	GPIO014	GPTP-IN6	PWM6	Reserved	
74	K12	GPIO015	0034	00000000	GPIO015	GPIO015	GPTP-OUT6	PWM7	Reserved	
75	F8	GPIO151	01A4	00000000	GPIO151	GPIO151	GPTP-IN3	ICT4	KSO15	
76	E9	GPIO152	01A8	00000000	GPIO152	GPIO152	GPTP-OUT3	ICT5	KSO16	
77	J9			N/A	VTR					
78	J10	GPIO003	000C	00000000	GPIO003	GPIO003	SMB00_DATA	Reserved	Reserved	
79	H10	GPIO004	0010	00000000	GPIO004	GPIO004	SMB00_CLK	Reserved	Reserved	
80	K11	GPIO005	0014	00000000	GPIO005	GPIO005	SMB01_DATA	Reserved	Reserved	
81	J12	GPIO006	0018	00000000	GPIO006	GPIO006	SMB01_CLK	Reserved	Reserved	
82	J11	GPIO012	0028	00000000	GPIO012	GPIO012	SMB07_DATA	SMB22_DATA	Reserved	
83	H11	GPIO013	002C	00000000	GPIO013	GPIO013	SMB07_CLK	SMB22_CLK	Reserved	
84	H12	GPIO130	0160	00000000	GPIO130	GPIO130	SMB12_DATA	Reserved	Reserved	
85	G10	GPIO131	0164	00000000	GPIO131	GPIO131	SMB12_CLK	Reserved	Reserved	
86	G11	GPIO132	0168	00000000	GPIO132	GPIO132	SMB06_DATA	KSO14	Reserved	
87	G12	GPIO140	0180	00000000	GPIO140	GPIO140	SMB06_CLK	Reserved	Reserved	
88	H9			N/A	VTR_FLASH					
89	G9	GPIO141	0184	00000000	GPIO141	GPIO141	SMB05_DATA	SMB20_DATA	FLSCLK	
90	H8	GPIO142	0188	00000000	GPIO142	GPIO142	SMB05_CLK	SMB20_CLK	FLSOUT	
91	G8	GPIO143	018C	00000000	GPIO143	GPIO143	SMB04_DATA	Reserved	FLSIN	
92	F9	GPIO144	0190	00000000	GPIO144	GPIO144	SMB04_CLK	Reserved	FLSCS	
93	F11	GPIO007	001C	00000000	GPIO007	GPIO007	SMB03_DATA	PS2_CLK0B	Reserved	
94	F10	GPIO010	0020	00000000	GPIO010	GPIO010	SMB03_CLK	PS2_DAT0B	Reserved	
95	F12	GPIO154	01B0	00000000	GPIO154	GPIO154	SMB02_DATA	PS2_CLK1B	Reserved	
96	E12	GPIO155	01B4	00000000	GPIO155	GPIO155	SMB02_CLK	PS2_DAT1B	Reserved	

TABLE 22-5: PIN CONTROL MUXING PIN REFERENCE NUMBERS [97:128]

Pin Ref. Number	Ball	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11	Notes
97	E11	GPIO110	0120	00000000	GPIO110	GPIO110	PS2_CLK2	GPTP-IN5	Reserved	
98	E10	GPIO111	0124	00000000	GPIO111	GPIO111	PS2_DAT2	GPTP-OUT5	Reserved	
99	D12	GPIO112	0128	00000000	GPIO112	GPIO112	PS2_CLK1A	KSO5	Reserved	
100	D11	GPIO113	012C	00000000	GPIO113	GPIO113	PS2_DAT1A	KSO6	Reserved	
101	C12	GPIO114	0130	00000000	GPIO114	GPIO114	PS2_CLK0A	Reserved	Reserved	
102	D10	GPIO115	0134	00000000	GPIO115	GPIO115	PS2_DAT0A	Reserved	Reserved	
103	C11	GPIO145	0194	00000000	GPIO145	GPIO145	SMB11_DATA	JTAG_TDI	Reserved	**Note 14**
104	B12	GPIO146	0198	00000000	GPIO146	GPIO146	SMB11_CLK	JTAG_TDO	Reserved	**Note 14**
105	B11	GPIO147	019C	00000000	GPIO147	GPIO147	SMB10_DATA	SMB21_DATA	JTAG_CLK	**Note 14**
106	A12	GPIO150	01A0	00000000	GPIO150	GPIO150	SMB10_CLK	SMB21_CLK	JTAG_TMS	**Note 14**
107	A11	None		N/A	JTAG_RST#	JTAG_RST#	Reserved	Reserved	Reserved	**Note 14**
108	C10	GPIO104	0110	00000000	GPIO104	GPIO104	UART_TX	Reserved	Reserved	
109	A10	GPIO105	0114	00000000	GPIO105	GPIO105	UART_RX	Reserved	Reserved	
110	B10	GPIO025	0054	00000000	GPIO025	GPIO025	UART_CLK	TIN0	EM_INT	
111	B9	GPIO026	0058	00000000	GPIO026	GPIO026	GPTP-IN0	TIN1	KSI3	
112	D9	GPIO027	005C	00000000	GPIO027	GPIO027	GPTP-OUT0	TIN2	KSI4	
113	C9	GPIO030	0060	00000000	GPIO030	GPIO030	GPTP-IN1	TIN3	KSI5	
114	A9	GPIO107	011C	00000000	GPIO107	GPIO107	KSO4	Reserved	Reserved	
115	D8	GPIO120	0140	00000000	GPIO120	GPIO120	KSO7	Reserved	Reserved	
116	B8	GPIO124	0150	00000000	GPIO124	GPIO124	GPTP-OUT4	KSO11	Reserved	
117	A8	GPIO125	0154	00000000	GPIO125	GPIO125	GPTP-IN4	KSO12	Reserved	
118	C8	GPIO031	0064	00000000	GPIO031	GPIO031	GPTP-OUT1	TOUT0	KSI6	
119	B7	GPIO032	0068	00000000	GPIO032	GPIO032	GPTP-IN2	TOUT1	KSI7	
120	D7	GPIO040	0080	00000000	GPIO040	GPIO040	GPTP-OUT2	TOUT2	KSO0	
121	A7	GPIO017	003C	00000000	GPIO017	GPIO017	GPTP-OUT7	TOUT3	KSI0	
122	F6	GPIO022	0048	00000000	GPIO022	GPIO022	BCM_B_CLK	V_CLK	Reserved	
123	E6	GPIO023	004C	00000000	GPIO023	GPIO023	BCM_B_DAT	V_DATA	Reserved	
124	F5	GPIO024	0050	00000000	GPIO024	GPIO024	BCM_B_INT#	V_FRAME	Reserved	
125	A6	GPIO045	0094	00000000	GPIO045	GPIO045	LSBCM_D_INT#	KSO1	Reserved	
126	C7	GPIO046	0098	00000000	GPIO046	GPIO046	LSBCM_D_DAT	KSO2	Reserved	
127	B6	GPIO047	009C	00000000	GPIO047	GPIO047	LSBCM_D_CLK	KSO3	Reserved	
128	C6	GPIO121	0144	00000000	GPIO121	GPIO121	BCM_A_INT#	KSO8	Reserved	

TABLE 22-6: PIN CONTROL MUXING PIN REFERENCE NUMBERS [129:144]

Pin Ref. Number	Ball	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11	Notes
129	A5	GPIO122	0148	00000000	GPIO122	GPIO122	BCM_A_DAT	KSO9	Reserved	
130	E5			N/A	VTR					
131	B5	GPIO123	014C	00000000	GPIO123	GPIO123	BCM_A_CLK	KSO10	Reserved	
132	A4	GPIO041	0084	00000000	GPIO041	GPIO041	PECI_REQUEST#	Reserved	Reserved	
133	A3	GPIO042	0088	00000000	GPIO042	GPIO042	BCM_C_INT#	PECI_DAT	SB-TSI_DAT	
134	A2	GPIO043	008C	00000000	GPIO043	GPIO043	BCM_C_DAT	PECI_RDY	SB-TSI_CLK	
135	A1	GPIO044	0090	00000000	GPIO044	GPIO044	BCM_C_CLK	VREF_PECI	Reserved	
136	B4	GPIO126	0158	00000000	GPIO126	GPIO126	KSO13	Reserved	Reserved	
137	D6	GPIO020	0040	00000000	GPIO020	GPIO020	KSI1	Reserved	Reserved	
138	B3	GPIO156	01B8	00000000	GPIO156	GPIO156	LED0	Reserved	Reserved	
139	B2	GPIO157	01BC	00000000	GPIO157	GPIO157	LED1	Reserved	Reserved	
140	B1	GPIO153	01AC	00000000	GPIO153	GPIO153	LED2	Reserved	Reserved	
141	D5			N/A	VSS					
142	C3			N/A	NO_CONNECT					
143	C5			N/A	NO_CONNECT					
144	E4			N/A	NO_CONNECT					

22.7 Power, Clocks and Reset

22.7.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

22.7.2 CLOCKS

This block uses the EC Bus Clock and the 64.52MHz MCLK. EC Bus Clock is used for access to registers. The MCLK is used for synchronizing the GPIO inputs.

22.7.3 RESET

This block is reset on a nSYS_RST. On reset, all Registers are reset to their default values.

22.8 Interrupts

Each pin in the GPIO Interface has both an interrupt and/or a Wake-up event. The interrupt source is routed onto the GPIO Status Bits corresponding to the specific GPIO identified in the GIRQ8 Source Register - GIRQ11 Source Register. The GPIO Interface can generate an interrupt on a high level, low level, rising edge and falling edge, as configured by the Interrupt Detection bits in the Pin Control Register associated with the GPIO signal function.

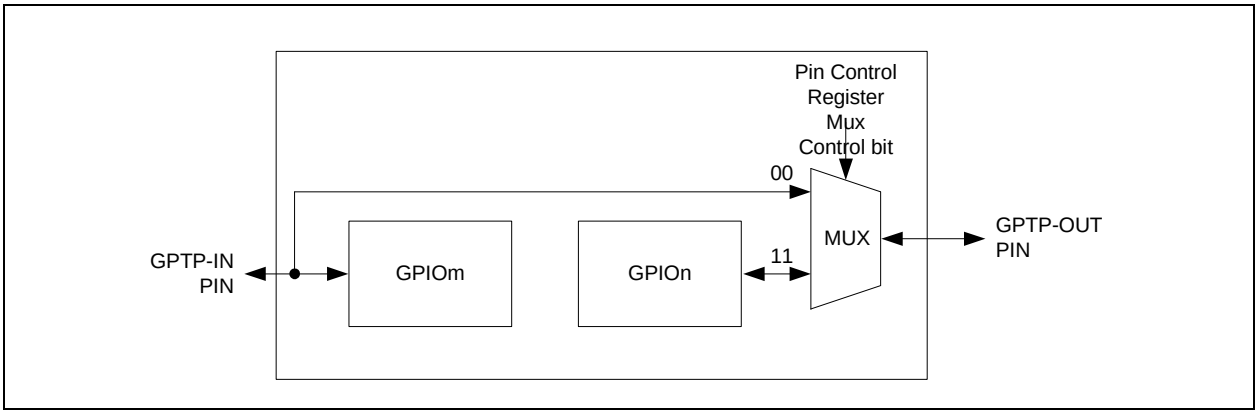
Note 22-1 The minimum pulse width ensured to generate an interrupt/wakeup event is 5ns.

22.9 GPIO Pass-Through Ports

GPIO Pass-Through Ports (GPTP) can multiplex two general purpose I/O pins as shown in Figure 22-3. GPIO Pass-Through Ports connect the GPTP-IN pin to the GPTP-OUT pin. The eight pin pairs are listed in Table 2.4.9, “General Purpose Pass-Through Ports Interface,” on page 17. The GPTP are sequentially assigned values 0:7. The GPTP port assignment have no relation to the GPIO Indexing assignments. The GPTP ports are controlled by the Mux Control bits in the Pin Control Register associated with the GPTP-OUT signal function.

In order to enable the GPTP Pass-Through Mode, the GPTP-IN (GPIOm in Figure 22-3) Pin Control Register must assign Mux Control = 00 (GPIO) and the GPIO Direction bit = 0 (input); the GPTP-OUT (GPIOn in Figure 22-3) Pin Control Register must assign Mux Control = the GPTP_OUT signal function and GPIO Direction bit = 1 (output). The Mux Control = GPTP-OUT signal function can differ from pin to pin See Section 22.4, "GPIO Indexing," on page 330.

FIGURE 22-3: GPIO PASS-THROUGH PORT EXAMPLE



Note 22-1 When Pass-Through Mode is enabled, the GPIOn output is disconnected from the GPIOn pin and the GPIOm pin signal appears on GPIOn pin. Note that in this case the GPIOm input register still reflects the state of the GPIOm pin.

22.10 Registers

The GPIO Interface Registers include GPIO Input Registers, GPIO Output Registers and a Pin Control Register for each signal function. The GPIO Interface has its own Logical Device Number and Base Address as indicated in Table 22-7.

Table 22-8 is a register summary for the GPIO Interface. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 22-7: GPIO Interface BASE ADDRESS TABLE

GPIOs **Blocks**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
GPIOs	31h	F0_C400h

TABLE 22-8: GPIO Interface REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Pin Control Register	000h - 200h	0-3	R/W	
GPIO[000:036] Output Register	280h	0-3	R/W	
GPIO[040:076] Output Register	284h	0-3	R/W	
GPIO[100:136] Output Register	288h	0-3	R/W	
GPIO[140:176] Output Register	28Ch	0-3	R/W	
GPIO[200:236] Output Register	290h	0-3	R/W	
GPIO[000:036] Input Register	300h	0-3	R/W	
GPIO[040:076] Input Register	304h	0-3	R/W	
GPIO[100:136] Input Register	308h	0-3	R/W	
GPIO[140:176] Input Register	30Ch	0-3	R/W	
GPIO[200:236] Input Register	310h	0-3	R/W	

APPLICATION NOTE: Bit31 in the five GPIO input registers (GPIO[200:236] Input Register, GPIO[140:176] Input Register, GPIO[100:136] Input Register, GPIO[040:076] Input Register, GPIO[000:036] Input Register) is a single bit register that can be set or cleared by software. It is provided to enable the use of the NORM instruction in the ARC instruction set in order to quickly find

the first set or cleared bit in the register. Setting Bit31 to '0b' makes the NORM instruction find the first set bit; setting Bit31 to '1b' makes the NORM instruction find the first cleared bit. For example, if Bit31 of GPIO[000:036] Input Register is '0b' and Bit17 is the highest numbered bit position for which a GPIO signal is set high, then the instruction sequence:

```
LD      R0,[GPIO[000:036] Input Register]
NORM    R0,R0
```

will return the value 17 in ARC register R0.

22.10.1 PIN CONTROL REGISTER

The Pin Control Register format is illustrated in Table 22-9 below and described in the subsections that follow. Pin Control Register address offsets and defaults are defined in Section 22.4, "GPIO Indexing," on page 330.

TABLE 22-9: PIN CONTROL REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336					32-bit	**EC SIZE**	
POWER	VTR			See Table 22-2 on page 333 – Table 22-6 on page 335			nSYS_RST **DEFAULT**	
	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**...**		**D26**	**D25**	**D24**
HOST TYPE	-							-
EC TYPE	R							R
BIT NAME	Reserved							GPIO input from pad
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-		-		-	-	-	-
EC TYPE	R	R	R		R	R	R/W	R/W
BIT NAME	Reserved						Alternative GPIO Write Enable	Alternative GPIO data
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-		-	-	-	-
EC TYPE	R		R/W		R/W	R/W	R/W	R/W
BIT NAME	Reserved		Mux Control		Polarity	Reserved	GPIO Direction	Output Buffer Type
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-				-	-	-	
EC TYPE	R/W				R		R/W	
BIT NAME	Interrupt Detection				Reserved		PU/PD	

22.10.1.1 PU/PD

TABLE 22-10: PU/PD BITS DEFINITION

Bit 1	Bit 0	Selected Function
0	0	None
0	1	Pull Up Enabled
1	0	Pull Down Enabled
1	1	None

22.10.1.2 Interrupt Detection

TABLE 22-11: Interrupt Detection BITS DEFINITION

D7	D6	D5	D4	Selected Function
0	1	X	X	Edge Detection power save mode (no clocks)
X Note 22-2	0	0	0	Low Level Sensitive
	0	0	1	High Level Sensitive
X	0	1	0	Reserved
	0	1	1	Reserved
1	1	0	0	Reserved
	1	0	1	Rising Edge Triggered
	1	1	0	Falling Edge Triggered
	1	1	1	Either edge triggered

Note 22-2 The most significant bit of the Interrupt Detection field controls edge detection power save mode. When this bit is set to ‘0’, edge triggered interrupts are blocked; however, level sensitive interrupts are forwarded to the interrupt block.

22.10.1.3 Output Buffer Type

TABLE 22-12: Output Buffer Type BIT DEFINITION

BIT 8	Selected Function
0	Push-Pull
1	Open Drain

Note 22-3 Unless explicitly stated otherwise, pins with (I/O/OD) or (O/OD) in their buffer type column in the tables in Section 2.5, "Pin Multiplexing," on page 21 are compliant with the following Programmable OD/PP Multiplexing Design Rule: Each compliant pin has a programmable open drain/push-pull buffer controlled by the Output Buffer Type bit in the associated Pin Control Register. The state of this bit controls the mode of the interface buffer for all selected functions, including the GPIO function.

22.10.1.4 GPIO Direction

The GPIO Direction bit controls the buffer direction only when the Mux Control field is ‘00’ selecting the pin signal function to be GPIO. When the Mux Control field is greater than ‘00’ (i.e., a non-GPIO signal function is selected) the GPIO Direction bit has no affect and the selected signal function logic directly controls the pin direction.

TABLE 22-13: GPIO Direction BIT DEFINITION

BIT 9	Selected Function
0	Input
1	Output

22.10.1.5 Polarity

When the Polarity bit is set to ‘1’ and the Mux Control bits are greater than ‘00,’ the selected signal function outputs are inverted and Interrupt Detection sense defined in Table 22-11, "Interrupt Detection Bits Definition" is inverted. When the Mux Control field selects the GPIO signal function (Mux = ‘00’), the Polarity bit does not effect the output. Regardless of the state of the Mux Control field and the Polarity bit, the state of the pin is always reported without inversion in the GPIO input register. See FIGURE 22-1: GPIO Interface Block Diagram on page 329.

TABLE 22-14: Polarity BIT DEFINITION

Bit 11	Description
0	Non-inverted
1	Inverted

22.10.1.6 Mux Control

TABLE 22-15: Mux Control BIT DEFINITION

BIT 13	BIT 12	Description
0	0	GPIO Function Selected
0	1	Signal Function 1 Selected
1	0	Signal Function 2 Selected
1	1	Signal Function 3 Selected

The Mux Control field determines the active signal function for a pin as defined in Table 22-15.

22.10.1.7 Alternative GPIO Write Enable

See description in Section 22.3 and Figure 22-2. Bit[17] defaults to ‘0’ on power-up.

TABLE 22-16: ALTERNATIVE GPIO BITS DEFINITION ON WRITES

BIT 17	BIT 16	Description
0	x	GPIO alternative write disabled
1	0	GPIO[x] out = ‘0’
1	1	GPIO[x] out = ‘1’

22.10.1.8 Alternative GPIO data

On writes, see Table 22-16. On reads, Bit [16] returns the programmed value.

22.10.1.9 GPIO input from pad

On reads, Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [17].

22.10.2 GPIO OUTPUT REGISTERS

22.10.2.1 Output GPIO[000:036]

TABLE 22-17: GPIO[000:036] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 and Table 22-8 on page 336				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 22-18 on page 340							
BIT NAME	GPIO[000:036] Output							

TABLE 22-18: BIT DEFINITIONS GPIO[000:036] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[007:000]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[017:010]	
[23:16]	R/W	GPIO[027:020]	
[26:24]	R/W	GPIO[032:030]	
[30:27]	R	GPIO[036:033]	Reserved
31	R	Reserved	This bit is always reserved

22.10.2.2 Output GPIO[040:076]

TABLE 22-19: GPIO[040:076] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336				32		**EC SIZE**	
POWER	VTR				0000_000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 22-20 on page 340							
BIT NAME	GPIO[040:076] Output							

TABLE 22-20: BIT DEFINITIONS GPIO[040:076] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[047:040]	Each bit monitors the corresponding pins status
[15:8]	R/W	GPIO[057:050]	
[23:16]	R/W	GPIO[067:060]	
[30:24]	R	GPIO[076:070]	Reserved
31	R	Reserved	This bit is always reserved

22.10.2.3 Output GPIO[100:136]

TABLE 22-21: GPIO[100:136] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 22-22 on page 341							
BIT NAME	GPIO[100:136] Output							

TABLE 22-22: BIT DEFINITIONS GPIO[100:136] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[107:100]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[117:110]	
[23:16]	R/W	GPIO[127:120]	
[26:24]	R/W	GPIO[132:130]	
[30:27]	R	GPIO[136:133]	Reserved
31	R	Reserved	This bit is always reserved

22.10.2.4 Output GPIO[140:176]

TABLE 22-23: GPIO[140:176] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 22-24 on page 341							
BIT NAME	GPIO[140:176] Output							

TABLE 22-24: BIT DEFINITIONS GPIO[140:176] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[147:140]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[157:150]	
[20:16]	R/W	GPIO[164:160]	
[23:21]	R	GPIO[167:165]	Reserved
[30:24]	R	GPIO[176:170]	
31	R	Reserved	This bit is always reserved

22.10.2.5 Output GPIO[200:236]

TABLE 22-25: GPIO[200:236] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336			32			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 22-26 on page 342							
BIT NAME	GPIO[200:236] Output							

TABLE 22-26: BIT DEFINITIONS GPIO[200:236] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[207:200]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[217:210]	
[23:16]	R	GPIO[227:220]	Reserved
[30:24]	R	GPIO[236:230]	
31	R	Reserved	This bit is always reserved

22.10.3 GPIO INPUT REGISTERS

22.10.3.1 Overview

The MEC1609/MEC1609i GPIO Interface includes four GPIO Input Registers which are always active as illustrated in FIGURE 22-1: GPIO Interface Block Diagram on page 329. The GPIO Input Registers can always be used to read the state of a pin (excluding the NORM Exception Bits), even when the pin is in an output mode and/or when a signal function other than the GPIO signal function is selected; i.e., the Pin Control Register Mux Control bits are not equal to '00.'

22.10.3.2 NORM Exception Bits

There can be up to 31 GPIO signal function names in each of the GPIO Input Registers (Bits D0 - D30). Bit D31 in each of these registers, the NORM Exception Bits, are single bit registers that can be set or cleared by software. The NORM Exception Bits are provided to enable the use of the NORM instruction in the ARC instruction set in order to quickly find the first set or cleared bit in the register. Setting Bit D31 to '0b' makes the NORM instruction find the first set bit; setting Bit D31 to '1b' makes the NORM instruction find the first cleared bit. For example, if Bit D31 of GPIO[000:036] Input Register is '0b' and Bit17 is the highest numbered bit position for which a GPIO signal is set high, then the following instruction sequence will return the value 17 in ARC register R0.

```
LD      R0,[GPIO[000:036] Input Register]
NORM    R0,R0
```

22.10.3.3 Input GPIO[000:036]

TABLE 22-27: GPIO[000:036] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336					32	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[000:036] INPUT						

22.10.3.4 Input GPIO[040:076]

TABLE 22-28: GPIO[040:076] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336					32	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[040:076] INPUT						

22.10.3.5 Input GPIO[100:136]

TABLE 22-29: GPIO[100:136] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336					32	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[100:136] INPUT						

22.10.3.6 Input GPIO[140:176]

TABLE 22-30: GPIO[140:176] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336			32			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[140:176] INPUT						

22.10.3.7 Input GPIO[200:236]

TABLE 22-31: GPIO[200:236] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 22-7 on page 336 - Table 22-8 on page 336			32			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[200:236] INPUT						

22.11 Programmer's Notes

As mentioned in Section 22.1 there are two ways to access GPIO input and output pin signals. This note aims to describe these access mechanisms and their relative merits. The two schemes can be employed concurrently.

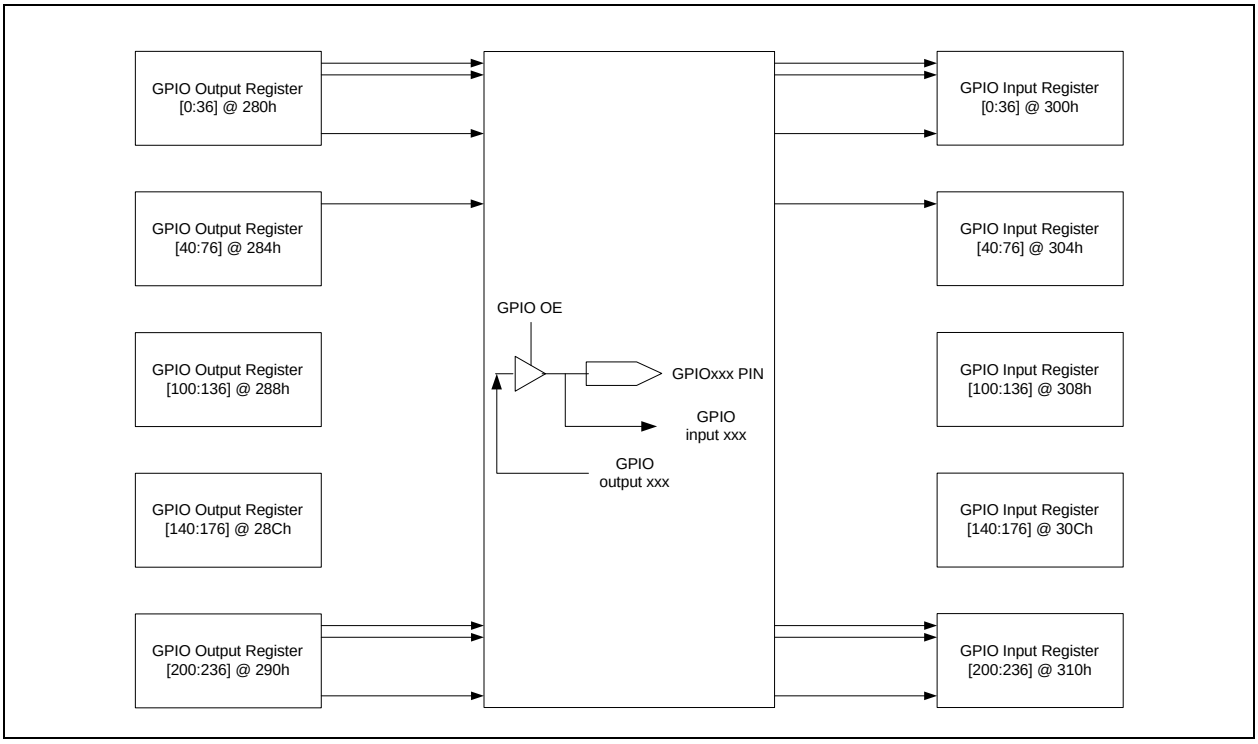
In the following description, the terms pins, ports, lines, signals are used interchangeably.

22.11.1 ACCESSING GPIO - MECHANISM 1

22.11.1.1 Overview

In the legacy approach, referred to hereafter as Mechanism 1, GPIO lines are grouped into 5 logical groups of 32 each. Associated with each group is an output register that controls the value to be driven out to the GPIO lines in the group and an input register whose value reflects the value seen on the group's GPIO lines' input side. Each non-reserved bit in the input/output register corresponds to a GPIO line. Since the number of GPIO lines that are implemented is less than that indicated by the logical labeling (0 to 236 in octal notation), some register bits are reserved. Figure 22-4 illustrates the arrangements.

FIGURE 22-4: ACCESSING GPIOS - SCHEME 1 (LEGACY SCHEME)



22.11.1.2 Access Mechanism

To change the output value of a particular GPIO line, the software first reads the GPIO Output Register associated with the group the line belongs to. The output register's value is logically (bitwise) ORed with a 32-bit value formed by setting the bit corresponding the GPIO line to the desired value while keeping all other bits zero. The result is then written to the same output register. Note that the value on GPIO output is as specified by its register bit value and is not affected by setting of Polarity in the Pin Control Register, which affects only the polarity of Alternate Function signal multiplexed on the GPIO pin. GPIO lines belonging to the same group can be changed together in one step.

To monitor the value on the input side of GPIO buffer, read the GPIO Input Register associated with the group the GPIO line belongs to. For example, GPIO_102 corresponds to Bit 2 in GPIO Input Register at offset 308h. Bit value can be extracted by applying appropriate mask and/or shift. In the current example, the mask would be 0x4; the shift, >>2.

Bit 31 in each GPIO Input Register, the NORM Exception Bits, is not associated with a GPIO line, rather it is to be used in conjunction with the ARC's NORM instruction to quickly locate the first bit in the register that is set or cleared. See Section 22.10.3.2 for details.

22.11.1.3 Applicability

Scheme 1 is simple and allows one to update multiple GPIO lines with one register update. However, when there are more than one software processes or threads accessing GPIO lines that share the same register, there are potential for l. In such cases the test-and-set sequence described in Section 22.11.1.2 must be an atomic operation. Since there is no support for this, software must implement a synchronization mechanism, e.g., mutex, semaphore, or spin lock, to serialize accesses.

22.11.2 ACCESSING GPIO - MECHANISM 2

22.11.2.1 Overview

In this mechanism each GPIO port is individually - as opposed to in group of 32 in Mechanism 1 -accessed via its Pin Control Register. The mechanism is enabled on a per-port basis. Once enabled, it overrides Mechanism 1, i.e., a write to a GPIO Output Register will not affect ports for which Mechanism 2 have been enabled. The mechanism can be dynamically enabled and disabled.

22.11.2.2 Access Mechanism

Figure 22-2 shows how relevant bits in the Pin Control Register are used to effect GPIO accesses in Mechanism 2.

To change a GPIO output port, the Alternative GPIO Write Enable bit its Pin Control Register is set to enable Mechanism 2, and the Alternative GPIO data bit is set to the port's desired output value. Both bits can be updated with a single register write. As long as Alternative GPIO Write Enable is set, Mechanism 2 is in effect. This bit must be cleared to switch to Mechanism 1.

The state of GPIO input port is reflected by the GPIO input from pad bit in the Pin Control Register. This holds true regardless of whether Mechanism 2 is enabled.

The port's input, output, and programmed output values are available in the same Pin Control Register.

22.11.2.3 Applicability

Mechanism 2 provides accesses at the port level and hence greatly reduces the unnecessary sharing of GPIO ports among different processes or threads. Even when a port is shared, the per-port control eliminates the need for atomic test-and-set.

Unless the applications call for the ability to update more than one GPIO port at the same time, it is recommended that Mechanism 2 be used. Alternatively, the two mechanisms can be together, with Mechanism 1 be applied to ports that do not need exclusive accesses.

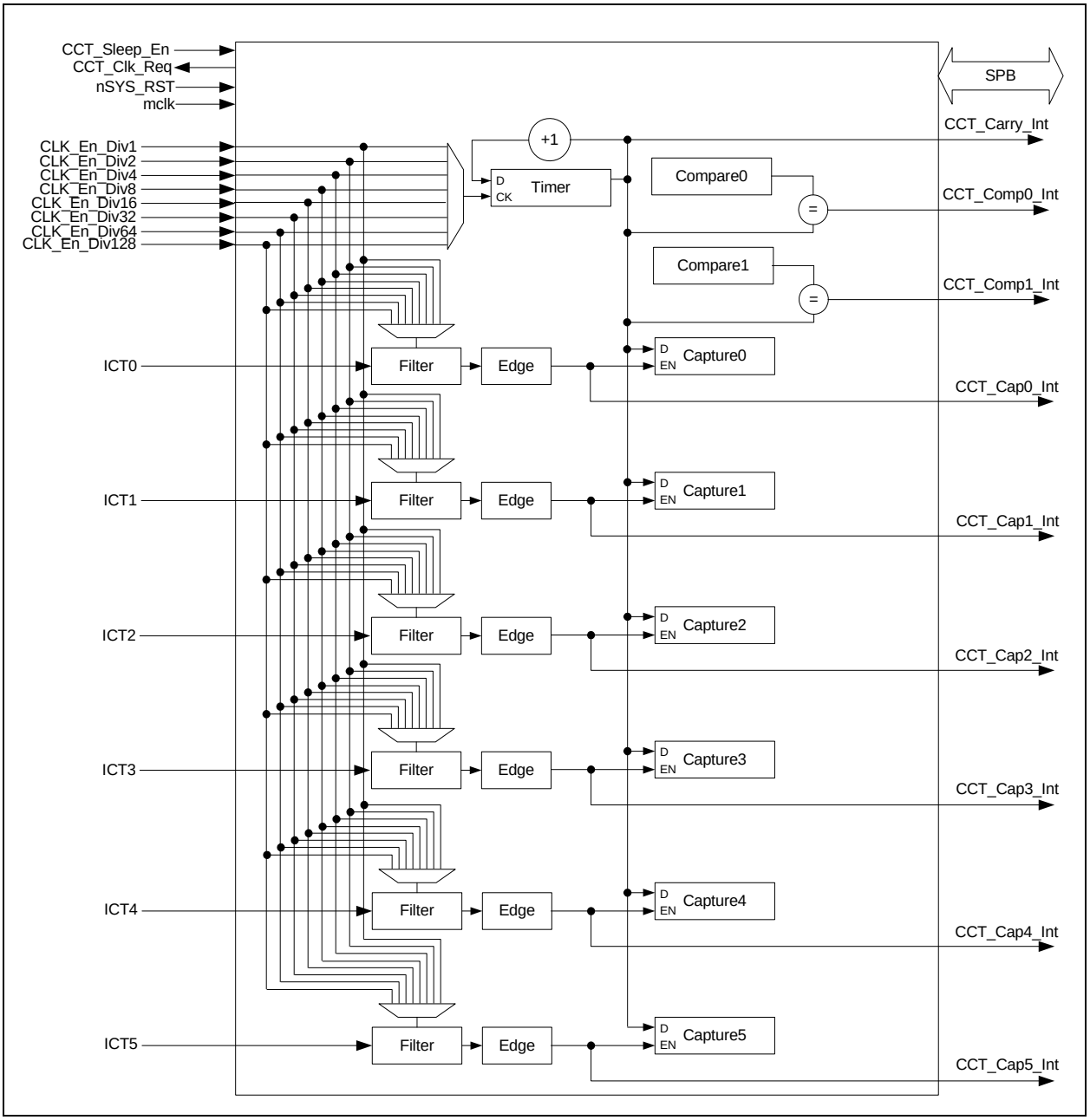
23.0 INPUT CAPTURE AND COMPARE TIMER

23.1 General Description

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

23.2 Capture and Compare Timer Block Diagram

FIGURE 23-1: CAPTURE AND COMPARE TIMER BLOCK DIAGRAM



23.3 Block Diagram Signal List

TABLE 23-1: Input Capture and Compare Timer **SIGNAL LIST**

Signal Name	Direction	Description
SPB	I/O Bus	MEC1609/MEC1609i peripheral bus
MCLK	INPUT	Master MEC1609/MEC1609i clock
nSYS_RST	INPUT	Block reset signal
CLK_EN_DIV1	INPUT	64.52MHz clock enable
CLK_EN_DIV2	INPUT	32.26MHz clock enable
CLK_EN_DIV4	INPUT	16.13MHz clock enable
CLK_EN_DIV8	INPUT	8.06MHz clock enable
CLK_EN_DIV16	INPUT	4.03MHz clock enable
CLK_EN_DIV32	INPUT	2.02MHz clock enable
CLK_EN_DIV64	INPUT	1.01MHz clock enable
CLK_EN_DIV128	INPUT	500KHz clock enable
ICT[5:0]	INPUT	External capture trigger signals (Note 23-1)
CCT_CARRY_INT	OUTPUT	Free-running timer wraparound interrupt
CCT_COMP[1:0]_INT	OUTPUT	Timer compare interrupts
CCT_CAP[5:0]_INT	OUTPUT	Timer capture interrupts
CCT_SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power Mode on page 349.
CCT_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= 64MHz can be turned 'off' when appropriate 1= 64MHz is required to be 'on.'

Note 23-1 External capture trigger signal inputs ICT0 - ICT3 are identical to FAN_TACH0 - FAN_TACH3 (see Section 2.4.7, "Fan Tachometer, PWM and Input Capture Timer Interface," on page 15). To use the Input Capture and Compare Timer for these pins, configure them for the Fan Tachometer functions as defined in Table 2-32, “Multiplexing Table (8 of 18),” on page 29 and Table 2-33, “Multiplexing Table (9 of 18),” on page 30.

External capture trigger signal inputs ICT4 - ICT5 are configured as defined in Table 2-34, “Multiplexing Table (10 of 18),” on page 31.

23.4 Power, Clocks and Reset

23.4.1 POWER DOMAIN

This block is powered by the VTR power supply.

23.4.2 CLOCKS

The timer in this unit is driven by mclk, the main system clock.

23.4.3 RESET

This block is reset on a nSYS_RST. On nSYS_RST the timer and all Capture and Compare registers are reset to their default values. The timer is also reset by the Free_Reset bit in the Capture and Compare Timer Control Register.

23.5 Interrupts

Interrupts from the Input Capture and Compare Timer block are routed to GIRQ23 Source Register of the Interrupt Aggregator. There are a total of nine interrupts from this block: one each for the six capture, one each for the two compare registers and one when the Free Running Timer wraps around. The interrupt signals are always generated by this block and can be queried or enabled through the Source and Enable registers in the Interrupt Aggregator.

23.6 Low Power Mode

This block is designed to conserve power when it is either sleeping or disabled. There are two ways to put the Capture and Compare block into a low power mode: Disable the Activate Bits or assert the CCT_SLEEP_EN signal to the Capture and Compare Timer block. The following table summarizes the Capture and Compare Timer behavior for each of these low power modes.

TABLE 23-2: BLOCK CLOCK GATING IN LOW POWER MODES

Activate Bits	CCT_SLEEP_En	Block Idle Status	CCT_CLK_REQ	State	Description
All 0	X	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.
Any 1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

23.7 Noise Filter

The noise filter uses Filter Clock (FCLK) to filter the signal on the ICTx pins. An external ICTx pin must remain in the same state for three FCLK ticks before the internal state changes. The Filter Bypass bit is used to bypass the noise filter. Each ICT input capture register can individually bypass the filter, but all ICT input capture registers that use the filter use the same Filter Clock.

- The signal ICT may be optionally only synchronized, or synchronized and filtered depending on the filter bypass bit.
- The minimum FCLK period must be at least 2X the duration of the ICT signal so that signal can be reliably captured in the bypass mode.
- The minimum FCLK period must be at least 4X the duration of the ICT signal so that signal can be reliably captured in the non-bypass mode.

23.8 Operation

23.8.1 INPUT CAPTURE

The Input Capture block consists of a free-running 32-bit timer and 6 capture registers. Each of the capture registers is associated with an input pin as well as an interrupt source bit in the Interrupt Aggregator: Table 23-3, "Pin Capture Interrupt Assignments" shows the assignment of pins to the Capture registers:

TABLE 23-3: PIN CAPTURE INTERRUPT ASSIGNMENTS

Capture Register	Capture Pin	Interrupt
Capture 0 Register	ICT0	CAPTURE 0, GIRQ23 Source Register
Capture 1 Register	ICT1	CAPTURE 1, GIRQ23 Source Register
Capture 2 Register	ICT2	CAPTURE 2, GIRQ23 Source Register
Capture 3 Register	ICT3	CAPTURE 3, GIRQ23 Source Register
Capture 4 Register	ICT4	CAPTURE 4, GIRQ23 Source Register
Capture 5 Register	ICT5	CAPTURE 5, GIRQ23 Source Register

The Capture registers store the current value of the Free Running timer whenever the associated input signal changes, according to the programmed edge detection. An interrupt is also generated to the EC. The Capture registers are read-only. The registers are updated every time an edge is detected. If software does not read the register before the next edge, the value is lost.

23.8.2 COMPARE INTERRUPT GENERATION

There are two 32-bit Compare registers. Each of these registers can independently generate an interrupt to the EC when the 32-bit free running Capture timer matches the contents of the Compare register.

23.9 Input Capture and Compare Timers Register Summary

There is one instance of the Input Capture and Compare Timer block implemented in the MEC1609/MEC1609i.

TABLE 23-4: Input Capture and Compare Timer BASE ADDRESS TABLE

Input Capture and Compare Timer **Instance**	**LDN**	**AHB Base Address**
Input Capture and Compare Timer	2h	F0_0800h

TABLE 23-5: Input Capture and Compare Timer REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Capture and Compare Timer Control Register	00h	3-0	R/W	
Capture Control 0 Register	04h	3-0	R/W	
Capture Control 1 Register	08h	3-0	R/W	
Free Running Timer Register	0Ch	3-0	R/W	
Capture 0 Register	10h	3-0	R	
Capture 1 Register	14h	3-0	R	
Capture 2 Register	18h	3-0	R	
Capture 3 Register	1Ch	3-0	R	
Capture 4 Register	20h	3-0	R	
Capture 5 Register	24h	3-0	R	
Compare 0 Register	28h	3-0	R/W	
Compare 1 Register	2Ch	3-0	R/W	

Note: The registers in this block with 32-bit values (Free Running Timer Register, Capture 0 Register, Capture 1 Register, Capture 2 Register, Capture 3 Register, Capture 4 Register, Capture 5 Register, Compare 0 Register, Compare 1 Register) should be read with 32-bit accesses. If these registers are read with multiple 8-bit or 16-bit accesses, the register values could change between the multiple accesses to the registers.

23.10 Detailed Register Descriptions

23.10.1 CAPTURE AND COMPARE TIMER CONTROL REGISTER

TABLE 23-6: CAPTURE AND COMPARE TIMER CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			32-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3-2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						Compare Enable1	Compare Enable0
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R	R/W	R/W	R/W
BIT NAME	Reserved	TCLK			Reserved	Free_ Reset	Free_ Enable	Activate

ACTIVATE

0=The timer block is powered down and all clocks are gated (default).

1=The timer block is in a running state.

FREE_ENABLE

Free-Running Timer Enable. This bit is used to start and stop the free running timer. This bit does not reset the timer count. The timer starts counting at 0000_0000h on reset and wraps around back to 0000_0000h after it reaches FFFF_FFFFh.

0=Timer is disabled. The Free Running Timer Register is writable.

1=Timer is enabled. The Free Running Timer Register is read-only.

Note:	The Free_Enable bit is cleared after the RESET cycle is done. Firmware must poll the Free_Reset bit to determine when it is safe to re-enable the timer.

FREE_RESET

Free Running Timer Reset. This bit stops the timer and resets the internal counter to 0000_0000h. This bit does not affect the Free_Enable bit. This bit is self clearing after the timer is reset.

0=Normal timer operation

1=Timer reset

TCLK

This 3-bit field sets the clock source for the Free Running Counter (see Section 23.10.4, "Free Running Timer Register," on page 356). The available frequencies are shown in Table 23-7:

TABLE 23-7: FREE RUNNING TIMER CLOCK FREQUENCIES

Timer Clock Select	Frequency Selected
000	64.52MHz
001	32.26MHz
010	16.13MHz
011	8.06MHz
100	4.03MHz
101	2.01MHz
110	1MHz
111	500KHz

COMPARE ENABLE0

Compare Enable for Compare 0 Register. If this bit is 1, a match between the Compare 0 Register and the Free Running Timer Register will cause an interrupt to be generated. If this bit is 0, no interrupt will be generated.

COMPARE ENABLE1

Compare Enable for Compare 1 Register. If this bit is 1, a match between the Compare 1 Register and the Free Running Timer Register will cause an interrupt to be generated. If this bit is 0, no interrupt will be generated.

23.10.2 CAPTURE CONTROL 0 REGISTER

TABLE 23-8: CAPTURE CONTROL 0 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL3			Reserved		Filter Byp3	Capture_Edge3	
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL2			Reserved		Filter Byp2	Capture_Edge2	
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL1			Reserved		Filter Byp1	Capture_Edge1	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL0			Reserved		Filter Byp0	Capture_Edge0	

CAPTURE_EDGE0

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 0 Register. See Table 23-9.

TABLE 23-9: CAPTURE EDGE SELECTION

Capture Edge Select	Edge that Triggers Capture
00	Falling edges
01	Rising edges
10	Both rising and falling edges
11	Capture event disabled

FILTER BYP0

Filter Bypass permits ICT0 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL0

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 23-10.

TABLE 23-10: FILTER CLOCK FREQUENCIES

Timer Clock Select	Frequency Selected
000	64.52MHz
001	32.26MHz
010	16.13MHz
011	8.06MHz
100	4.03MHz
101	2.01MHz
110	1MHz
111	500KHz

CAPTURE_EDGE1

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 1 Register. See Table 23-9.

FILTER BYP1

Filter Bypass permits ICT1 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL1

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 23-10.

CAPTURE_EDGE2

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 2 Register. See Table 23-9.

FILTER BYP2

Filter Bypass permits ICT2 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL2

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 23-10.

CAPTURE_EDGE3

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 3 Register. See Table 23-9.

FILTER BYP3

Filter Bypass permits ICT3 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL3

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 23-10.

23.10.3 CAPTURE CONTROL 1 REGISTER

TABLE 23-11: CAPTURE CONTROL 1 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			32-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3-2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL5			Reserved		Filter Byp5	Capture_Edge5	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL4			Reserved		Filter Byp4	Capture_Edge4	

CAPTURE_EDGE4

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 4 Register. See Table 23-9.

FILTER BYP4

Filter Bypass permits ICT4 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL4

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 23-10.

CAPTURE_EDGE5

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 5 Register. See Table 23-9.

FILTER BYP5

Filter Bypass permits ICT5 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL5

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 23-10.

23.10.4 FREE RUNNING TIMER REGISTER

TABLE 23-12: FREE RUNNING TIMER REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	0Ch				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Free Running Timer[31:0]							

FREE RUNNING TIMER[31:0]

This register contains the current value of the Free Running Timer. A Capture Timer interrupt is signaled to the Interrupt Aggregator when this register transitions from FFFF_FFFFh to 0000_0000h.

When Free_Enable in Capture and Compare Timer Control Register is 1, this register is read-only. When Free_Enable is 0, this register may be written.

23.10.5 CAPTURE 0 REGISTER

TABLE 23-13: CAPTURE 0 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	10h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 0[31:0]							

CAPTURE 0[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT0.

23.10.6 CAPTURE 1 REGISTER

TABLE 23-14: CAPTURE 1 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	14h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 1[31:0]							

CAPTURE 1[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT1.

23.10.7 CAPTURE 2 REGISTER

TABLE 23-15: CAPTURE 2 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	18h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 2[31:0]							

CAPTURE 2[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT2.

23.10.8 CAPTURE 3 REGISTER

TABLE 23-16: CAPTURE 3 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	1Ch				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 3[31:0]							

CAPTURE 3[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT3.

23.10.9 CAPTURE 4 REGISTER

TABLE 23-17: CAPTURE 4 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	20h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 4[31:0]							

CAPTURE 4[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT4.

23.10.10 CAPTURE 5 REGISTER

TABLE 23-18: CAPTURE 5 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	24h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 5[31:0]							

CAPTURE 5[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT5.

23.10.11 COMPARE 0 REGISTER

TABLE 23-19: COMPARE 0 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
BUS	EC SPB							
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Compare 0[31:0]							

COMPARE 0[31:0]

A Compare 0 interrupt is generated when this register matches the value in the Free Running Timer.

23.10.12 COMPARE 1 REGISTER

TABLE 23-20: COMPARE 1 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	2Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Compare 1[31:0]							

COMPARE 1[31:0]

A Compare 1 interrupt is generated when this register matches the value in the Free Running Timer.

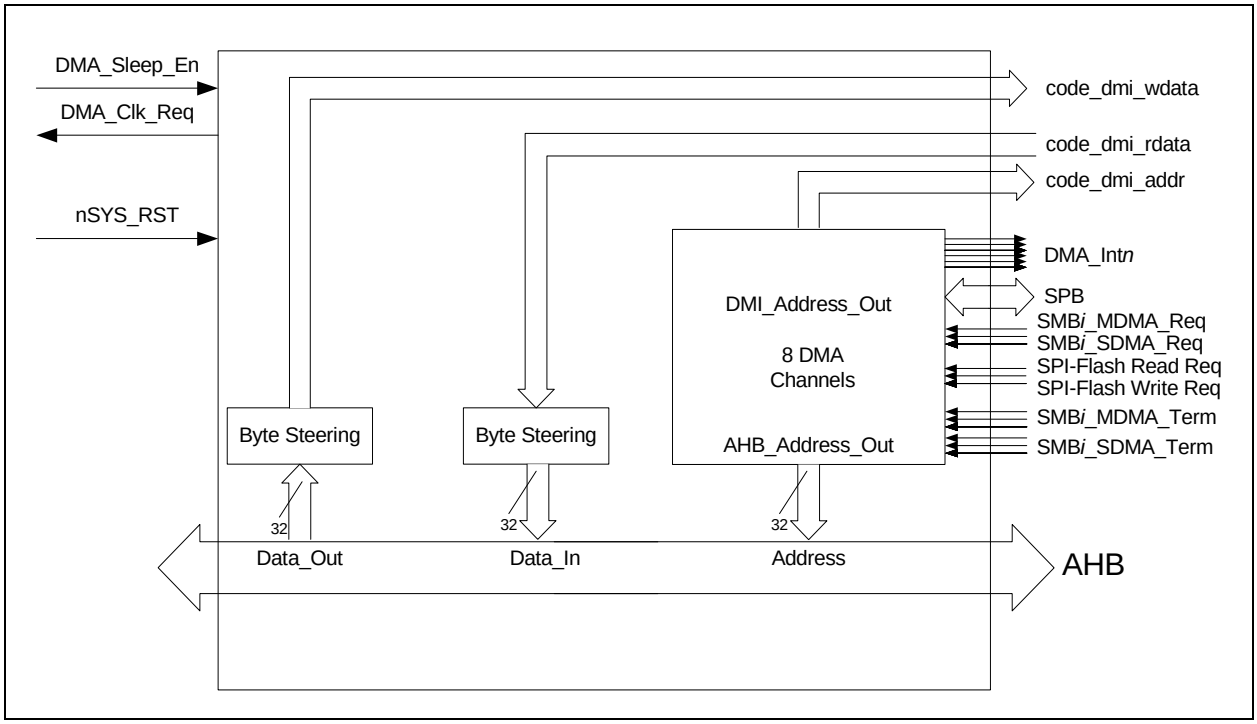
24.0 DMA CONTROLLER

24.1 General Description

This block describes the MEC1609/MEC1609i DMA controller. The DMA controller is designed to move data between the SMBus and SPI Flash controllers and the EC closely-coupled SRAM memory. There are eight independent channels that each move byte-wide data between the SMBus and SPI Flash controllers and the SRAM in either direction.

24.2 DMA Block Diagram

FIGURE 24-1: DMA BLOCK DIAGRAM



24.3 Block Diagram Signal List

TABLE 24-1: DMA Controller SIGNAL LIST

Signal Name	Direction	Description
AHB	I/O Bus	MEC1609/MEC1609i system bus
SPB	I/O Bus	MEC1609/MEC1609i peripheral bus
code_dmi_wdata, code_dmi_rdata, code_dmi_addr	I/O Bus	Direct Memory Interface (DMI) to ARC ICCM memory
MCLK	INPUT	Master MEC1609/MEC1609i clock
nSYS_RST	INPUT	Block reset signal
DMA_Int[7:0]	OUTPUT	DMA Interrupt signals
SMB[2:0]_MDMA_Req	INPUT	DMA request control from SMBus Master channel.
SMB[2:0]_SDMA_Req	INPUT	DMA request control from SMBus Slave channel.
SMB[2:0]_MDMA_Term	INPUT	DMA termination control from SMBus Master channel.
SMB[2:0]_SDMA_Term	INPUT	DMA termination control from SMBus Slave channel.
SPIFLASH_Read_Req	INPUT	DMA request control from SPI Flash receive Buffer register

TABLE 24-1: DMA Controller SIGNAL LIST (CONTINUED)

Signal Name	Direction	Description
SPIFLASH_Write_Req	INPUT	DMA request control from SPI Flash transmit Buffer register
DMA_SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power Mode on page 361.
DMA_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= 64MHz can be turned 'off' when appropriate 1= 64MHz is required to be 'on.'

24.4 Power, Clocks and Reset

24.4.1 POWER DOMAIN

This block is powered by the VTR power supply with a separate Analog supply (AVDD).

24.4.2 CLOCKS

This block has three clock inputs: MCLK at 64.52 MHz, the AHB bus clock enable, used for AHB transfers, and the EC clock enable, used for DMI transfers.

APPLICATION NOTE: The DMA Controller requires the EC clock in order to write into the EC SRAM. The EC, therefore, must not be in sleep mode at any time during a DMA transfer.

24.4.3 RESET

This block is reset on a nSYS_RST.

24.5 DMA Interrupts

Each channel of the DMA controller generates an interrupt event to the EC which indicate a DMA transfer is complete.

24.6 Low Power Mode

This block is designed to conserve power when it is either sleeping or disabled. There are two ways to put the DMA Controller into a low power mode: Disable all DMA channels via the Activate Bits or assert the DMA_SLEEP_EN signal to the DMA Controller. The following table summarizes the DMA Controller behavior for each of these low power modes.

TABLE 24-2: BLOCK CLOCK GATING IN LOW POWER MODES

Activate	DMA_SLEEP_EN	Busy	DMA_CLK_REQ	State	Description
All 0	X	X	0	INACTIVE	All channels are disabled
Any 1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
Any 1	1	1	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
Any 1	1	0	0	SLEEPING	The block is idle and the core clock can be stopped.

24.7 Operation

The MEC1609/MEC1609i features a eight channel DMA controller. The DMA controller can autonomously move data from I/O devices to and from EC local memory without EC intervention.

The DMA has the following characteristics:

- Data is only moved 1 byte (8 bits) at a time
- Data only moves between devices on the AHB bus, or devices connected to an AHB bus bridge, and the EC

SRAM. Since the SRAM is dual-ported to be both ICCM and DCCM, the DMA can be interpreted as moving data into and out of the DCCM as well as the ICCM.

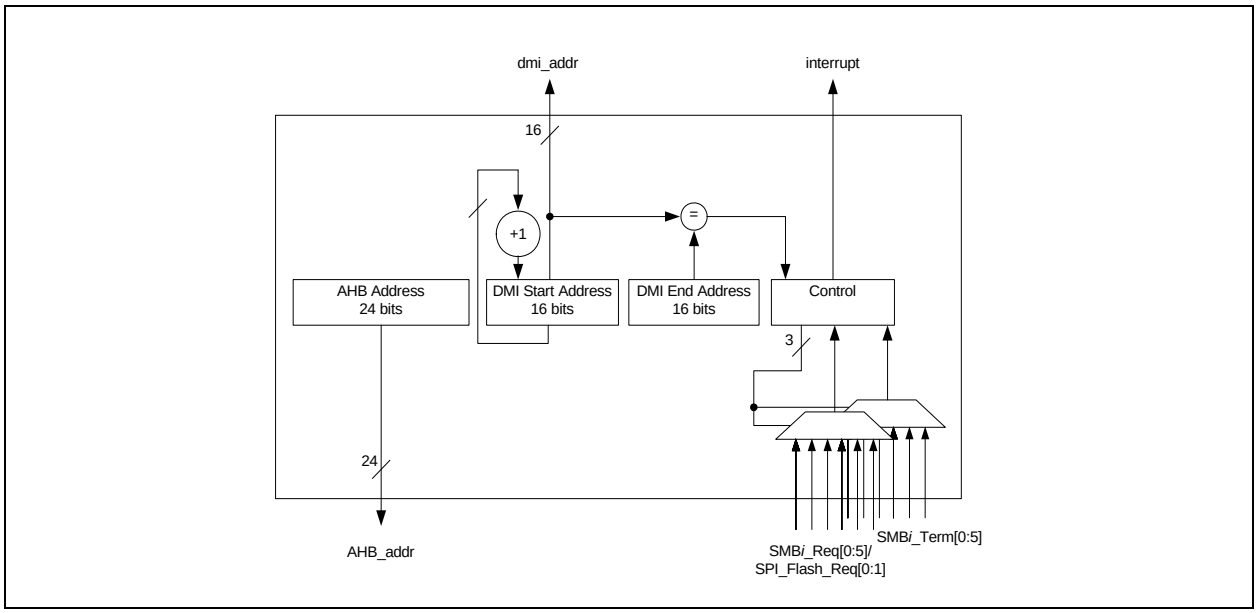
- The number of DMA channels can be less than the number of I/O devices that can use the DMA for data transfers, so the DMA channels are shareable and can be assigned to any device.

The DMA controller is not designed to communicate with I/O devices with more than an 8-bit interface. The controller will accesses SRAM buffers only with incrementing addresses (that is, it cannot start at the top of a buffer, nor does it handle circular buffers automatically). The controller does not handle chaining (that is, automatically starting a new DMA transfer when one finishes).

24.7.1 DMA CHANNELS

Each DMA channel is capable of bi-directional data movement between an logical device and the EC closely-coupled memory. A single DMA channel is illustrated in Figure 24-2, "DMA Channel":

FIGURE 24-2: DMA CHANNEL



There are 8 possible logical devices that may connect to a DMA Controller channel. There are three SMBus controllers, each of which has a separate read request and write request. In addition, the SPI Flash controller has a read request and a write request. The SMBus controllers also provide a termination signal for each direction of each device. The SPI Flash device does not provide termination signals, so the two termination inputs corresponding to the SP Flash Read request and the SPI Flash Write request are always held to 0.

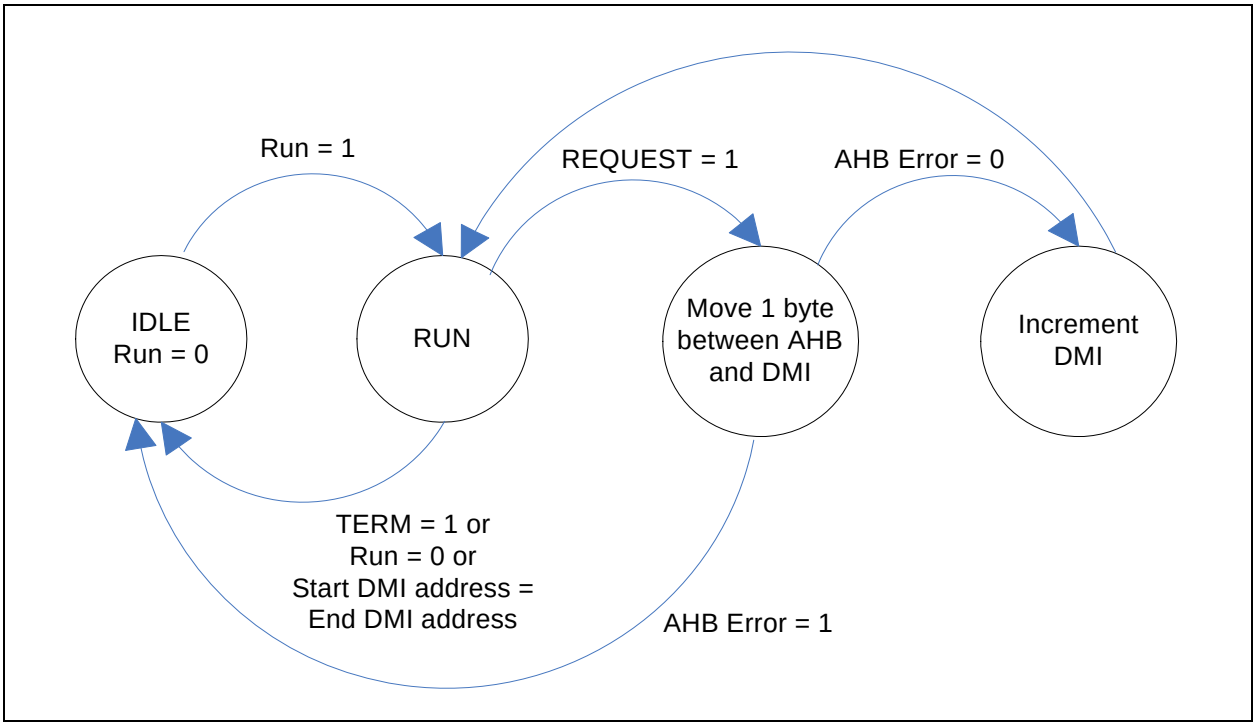
Based on the DIR in the channel's DMA Control Register, bytes are copied from a device's Receive Buffer to the ICCM/DCCM, or from the ICCM/DCCM to a device's Transmit Buffer. The AHB Address Register is programmed to be the address of the buffer required for the transfer. Software is responsible for insuring that this address is associated with the correct SRAM buffer defined by the DMI Start Address Register and by the correct SMBus as selected by the DEVICE field in the Control register.

The DMI Start Address is an offset from the base of the SRAM. It is an offset from both the base of the DCCM and the base of the ICCM, since the two memories are different aliases of a single dual-ported SRAM. This register is loaded into a 16-bit counter, which increments by 1 under state machine control in order to generate the current DMI address.

The End Address register contains the address one greater than the last byte to transfer. The DMA transfer terminates when the current DMI address equals the End Address. If the DMA channel is configured with the Start Address and End Address registers set to the same address, no bytes will be transferred. If the End Address is configured with an address that is less than the Start Address, the DMI Start Address register will wrap around from FFFFh to 0000h. The DMA Controller always sends 16 address bits to the SRAM. If fewer than 16 bits are required to address the ICCM/DCCM, then the SRAM will ignore the upper bits of the Start Address and End Address registers. For example, if the ICCM/DCCM is 16KB, bits 14 to 15 in the Start and End Address registers will be ignored by the SRAM.

The state machine that runs each DMA channel is illustrated in Figure 24-3, "Channel State Machine". When software sets the bit to 1, the state machine enters the RUN state and waits for an assertion of the selected REQUEST input. As long as REQUEST is asserted when the state machine is in the RUN state the DMA controller starts a 1-byte wide AHB bus transaction, in the direction defined by DIR. After the AHB transaction completes, the current DMI address counter is incremented by 1 and compared to the End Address register. If the current address is not equal to the End Address, the state machine returns to the RUN state. If the current address is equal to the End Address, or if the TERM input is asserted, the DMA transaction is terminated and the state machine returns to the IDLE state. If there is an AHB bus error on the transfer between the AHB and the DMI, the DMI address counter increment is inhibited. The state machine returns to the IDLE state and the Status is set to AHB Bus Error.

FIGURE 24-3: CHANNEL STATE MACHINE



24.7.2 I/O DEVICES

The DMA Controller is configured to work with any of the three SMBus controllers and the SPI Flash controller in the MEC1609/MEC1609i. Table 24-3, "DMA Device Selection" shows the mapping between the AHB Address and DEVICE and DIR fields for each I/O device.

TABLE 24-3: DMA DEVICE SELECTION

DEVICE	DIR	Device Name	Data AHB Address	Request Signals
0	0	SMBUS Controller 0	F0_184Ch (SMBus Slave Receive Buffer)	SMB0_SDMA_Req
0	1		F0_1848h (SMBus Slave Transmit Buffer)	
1	0		F0_1854h (SMBus Master Receive Buffer)	SMB0_MDMA_Req
1	1		F0_1850h (SMBus Master Transmit Buffer)	

TABLE 24-3: DMA DEVICE SELECTION (CONTINUED)

DEVICE	DIR	Device Name	Data AHB Address	Request Signals
2	0	SMBUS Controller 1	F0_18CCh (SMBus Slave Receive Buffer)	SMB1_SDMA_Req
2	1		F0_18C8h (SMBus Slave Transmit Buffer)	
3	0		F0_18D4h (SMBus Master Receive Buffer)	SMB1_MDMA_Req
3	1		F0_18D0h (SMBus Master Transmit Buffer)	
4	0	SMBUS Controller 2	F0_194Ch (SMBus Slave Receive Buffer)	SMB2_SDMA_Req
4	1		F0_1948h (SMBus Slave Transmit Buffer)	
5	0		F0_1954h (SMBus Master Receive Buffer)	SMB2_MDMA_Req
5	1		F0_1950h (SMBus Master Transmit Buffer)	
6	0	SPI Flash Controller	FF_3C10h (SPI RX_Data Register)	SPIFLASH_Read_Req
6	1		FF_3C0Ch (SPI TX_Data Register)	SPIFLASH_Write_Req
7	0/1	Reserved	–	–

Note 24-1 See the DEVICE field in the DMA Control Register on page 368.

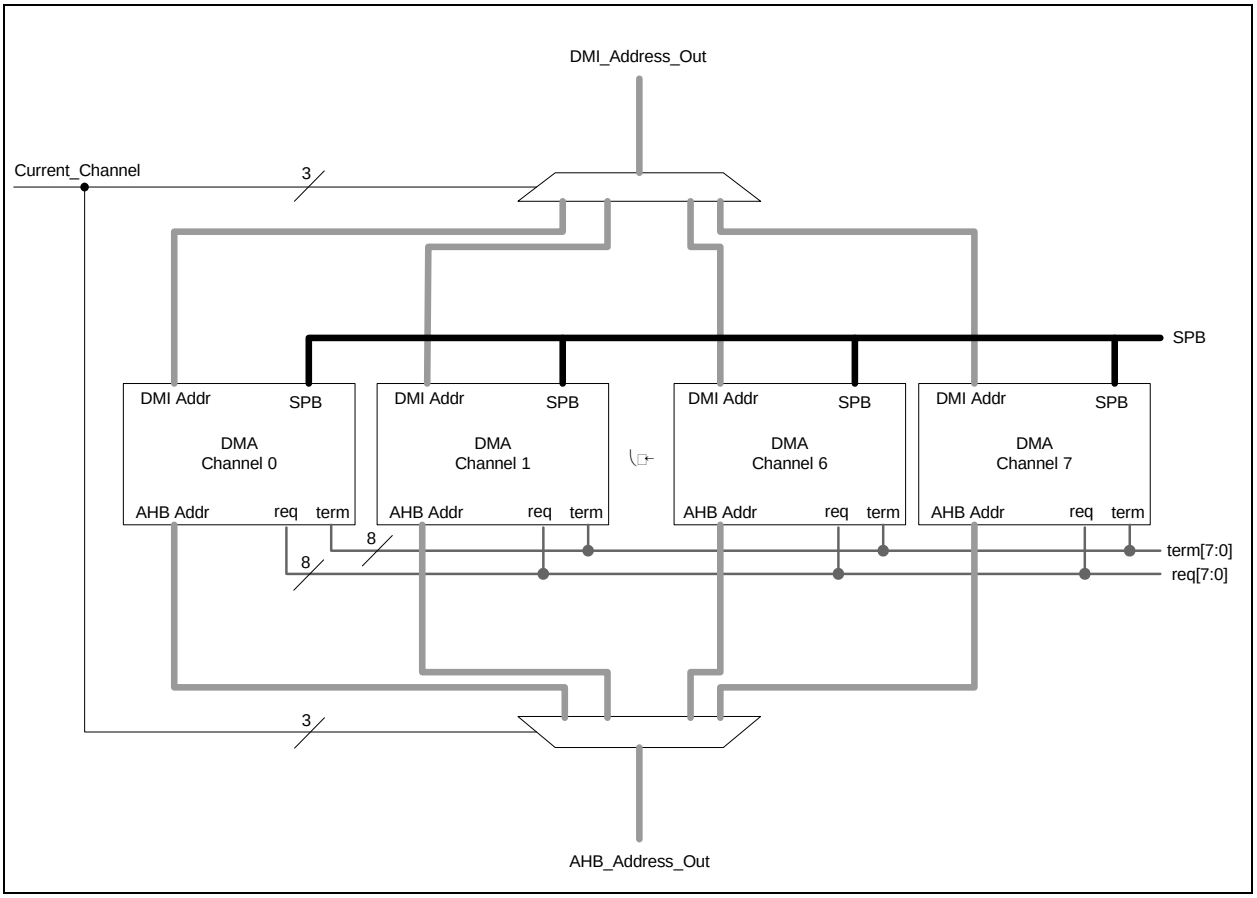
Note 24-2 See the DIR bit in the DMA Control Register on page on page 368.

The Request signals from the devices are the Data register status signals. In the read (0, from the device to SRAM) direction, a DMA request is asserted as long as the Receive Data register is not empty. In the transmit (1, from the SRAM to the device) direction, a DMA request is asserted as long as the Transmit Data register is not full. For the SMBus controllers, the Terminate signal is asserted if the SMBus controller detects an error condition during an SMBus transaction, or if software shuts down the SMBus controller. There is no Terminate signal for the SPI Flash controller.

24.7.3 DMA CHANNEL ARBITRATION AND MULTIPLEXING

The eight DMA channels share the DMI interface and the AHB bus. Figure 24-4, "DMA Channel Multiplexing" illustrates the multiplexing of the channels onto the busses.

FIGURE 24-4: DMA CHANNEL MULTIPLEXING



A DMA Channel is ready to run as long as it is in the Run state and its selected Request input is asserted. The DMA controller services DMA Channels on a first-come first-served basis. If two channels become ready to run simultaneously, they are served in numerical order (channel 0 before channel 1, etc.). If multiple channels are continuously ready (that is, their respective Request inputs are always asserted), then they will be served in round-robin order.

24.8 DMA Registers

The base address for the DMA Controller block in the AHB address space is listed in Table 24-4, "DMA Controller Base Address Table".

TABLE 24-4: DMA Controller BASE ADDRESS TABLE

DMA Controller Instance	LDN from (Table 3-2 on page 48)	AHB Base Address
DMA Channel0	9h	F0_2400h + 000h
DMA Channel1		F0_2400h + 020h
DMA Channel2		F0_2400h + 040h
DMA Channel3		F0_2400h + 060h
DMA Channel4		F0_2400h + 080h
DMA Channel5		F0_2400h + 0A0h
DMA Channel6		F0_2400h + 0C0h
DMA Channel7		F0_2400h + 0E0h

The following table summarizes the registers allocated for the DMA Controller. The offset field in the following table is the offset from the AHB Base Address defined in Table 24-4 on page 366.

TABLE 24-5: DMA Controller REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
DMA Control Register	0h	1-0	R/W	
DMI End Address Register	4h	1-0	R/W	
DMI Start Address Register	8h	1-0	R/W	
AHB Address Register	Ch	3-0	R/W	
DMA Activate Register	10h	0	R/W	

24.8.1 DMA CONTROL REGISTER

The DMA Control Register is used to control the behavior of the DMA controller.

TABLE 24-6: DMA CONTROL REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				Device			DIR
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved		Busy	Status		Done	Request	Run

RUN

When this bit is 1, the state machine is active and the channel continually tries to move a byte between the I/O device at AHB Address and SRAM at DMA Address. When this bit is 1, software cannot modify any of the other registers in the channel, or any other bits in this register besides , in order to insure that no AHB transaction is modified while it is in progress.

Setting this bit to 0 will halt the DMA function. If there is an AHB transfer in progress when this bit is set to 0 the transfer will complete before the DMA state machine returns to the IDLE state. Firmware should query the Busy bit after setting Run to 0 in order to determine when the DMA transaction has terminated.

The DMA_Int signal is only asserted when is 1.

REQUEST

Read-only. This bit is always 0 when Run is 0, and is set when the DMA request input is 1. The DMA request input is selected by DEVICE and DIR.

DONE

Read-only. This bit is always 0 when is Run is 0, and is 1 when the DMA Controller state machine returns to the IDLE state. The DMA Controller state machine will transition back to IDLE when DMI Start Address Register equals the DMI End Address Register, when the DMA Termination input is 1 or if the AHB transaction is terminated with a bus error. The DMA Termination input is selected by DEVICE and DIR. This bit is routed to the interrupt controller.

STATUS

Read-only. This field is updated whenever Done goes from 0 to 1 or when Run goes from 1 to 0, and indicates why a DMA transfer completed. Status values are:

00: Run is set to 0. This field is always 0 when Run is 0.

01: Start Address matched End Address

10: DMA_Term input asserted

11: An AHB bus error occurred on the transfer

Status values are shown in order of priority. If more than one condition caused a return to the IDLE state, the condition with the lowest Status value is reported.

BUSY

Read-only. This bit is 1 when the DMA State Machine is not in the IDLE state and 0 when the DMA State Machine is in the IDLE state.

DIR

DMA transfer direction. 0 for reads from the AHB device to DCCM memory, 1 for writes from DCCM memory to the AHB device. When combined with DEVICE, determines which DMA_request input is used to start a DMA transfer

DEVICE

The DEVICE field selects which I/O device is assigned to this DMA channel. See Table 24-3, “DMA Device Selection,” on page 363 for the DEVICE field to Device Name mapping.

24.8.2 DMI END ADDRESS REGISTER

This address defines the DMA stops transferring. When the incrementer that was loaded from the DMI Start Address Register is equal to this register, the DMA completes.

TABLE 24-7: DMI END ADDRESS REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	04h				16-bit		**EC SIZE**	
POWER	VTR				00h		**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_End_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_End_Address[7:0]							

DMI_END_ADDRESS

This field contains the address one past the last byte to be transferred for the DMA channel. The DMA transfer stops when the current DMI address is equal to this register. If the End Address register is equal to the Start Address register when is set to 1, no data are transferred and the DMA terminates immediately.

24.8.3 DMI START ADDRESS REGISTER

Note: This register is 16-bit only. It does not support 8-bit accesses.

TABLE 24-8: DMI START ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	08h			16bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_Start_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_Start_Address[7:0]							

DMI_START_ADDRESS[15:0]

This field defines an offset from the base of the SRAM, which is an offset from both the base of the DCCM and the base of the ICCM, since the two memories are different aliases of a single dual-ported SRAM. This register is loaded into a 16-bit counter, which increments by 1 under state machine control, when is set to 1. This register defines the initial byte address for bytes to be transferred on the associated DMA channel.

When first written by software, this register contains the start address in the DCCM for the DMA transfer. While a DMA transfer is in progress, this register contains the address of the next byte to be transferred. When the DMA transfer completes, this register is one greater than the address of the last byte transferred. Software can determine how many bytes were transferred overall by subtracting the value it used to configure this register initially from the value of this register when the transfer completes.

24.8.4 AHB ADDRESS REGISTER

The AHB Address Register is the address of the I/O device that is the source or sink of the DMA transfer.

TABLE 24-9: AHB ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	0Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AHB_Address[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AHB_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AHB_Address[7:0]							

AHB_ADDRESS[23:0]

This is the address of the I/O port in the AHB address space. Software is responsible for insuring that this address is the correct address for the I/O device assigned to the channel.

24.8.5 DMA ACTIVATE REGISTER

The DMA Activate Register is used to gate clocks to a DMA channel, in order to conserve power. Software must set the Activate bit to ‘1b’ in order for a channel to operate.

TABLE 24-10: DMA ACTIVATE REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	10h			8-bit			**EC SIZE**	
POWER	VTR			00h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 0, the MCLK is gated to this channel, so the channel will not operate. When this bit is 1, the channel is provided with the system clock and the channel can operate.

25.0 SMB DEVICE INTERFACE

25.1 General Description

The MEC1609/MEC1609i SMB Device Interface includes three instances of an SMBus controller core: SMBus 0, SMBus 1 and SMBus 2. This chapter describes aspects of the SMB Device Interface that are unique to the MEC1609/MEC1609i instantiations of this core; including, Power Domain, Resets, Clocks, Interrupts, Registers and the Physical Interface. For a *General Description*, *Features*, *Block Diagram*, *Functional Description, Registers Interface* and other core-specific details, see Ref [1] (note: in this chapter, *italicized text* typically refers to SMBus controller core interface elements as described in Ref [1]).

25.1.1 REFERENCES

1. SMBus Controller Core Interface, Revision 2.0, v2.17, Core-Level Architecture Specification, 2/11/09

25.1.2 SMB PIN SIGNAL INTERFACE DESCRIPTION

The pin signals are defined in Table 2.4.13, “SMBus Interface,” on page 19.

25.2 Power, Clocks and Reset

25.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains. For more detail about the SMBus controller core Power Domain, see *Section 3.1, “Power Configuration”* in Ref [1].

25.2.2 CLOCKS

SMB Device Interface Clocking is described below in Table 25-1. Use this table when programming the SMBus controller core bus clock and timing values as specified in Ref [1].

TABLE 25-1: SMB Device Interface CLOCKING

Clock Source (Note 25-1)	SMBus Controller Core Clock (Note 25-2)	Frequency	Description
MCLK	CORE_CLK	64.52 MHz	–
MCLK_DIV8_EN	BAUD_CLK_EN	8.06 MHz	Use this frequency when programming the *Bus Clock Register*, *Data Timing Register* and the *Time-Out Scaling Register* described in Ref [1].
EC_BUS_CLK_EN	SPB_CLK_EN	Programmable	EC Bus Clock.

Note 25-1 See Section 5.4.8, "Ring Oscillator Sourced Clocking," on page 91.

Note 25-2 For more detail about SMBus controller core Clocking see *Chapter 2, “Hardware Interface”* and *Section 3.3,* “*Clocking*” in Ref [1].

25.2.3 RESETS

Each of the SMBus controller core instances in the MEC1609/MEC1609i SMB Device Interface are reset by nSYS_RST. See Section 5.6, "Reset Interface," on page 95 for details on resets in the MEC1609/MEC1609i. For more detail about SMBus controller core Resets, see *Section 3.2,* “*Reset Interface*” in Ref [1].

25.3 Interrupts

Each EC SMB Controller has both an activity interrupt event and a START Bit detection Wake-up event. The SMB activity interrupt events are routed to the SMB0, SMB1 and SMB2 bits in the GIRQ12 Source Register on page 270. The START Bit detection Wake-up events are routed to the SMB00 WK, SMB01 WK, SMB02 WK, SMB03 WK, SMB04 WK, SMB05 WK, SMB06 WK, SMB07 WK, SMB10 WK, SMB11 WK, SMB12 WK, SMB20 WK, SMB21 WK, SMB22 WK and SB_TSI bits in the GIRQ12 Source Register on page 270. The edge detection of the interrupt and wake events are controlled by their associated pin control registers in the Section 22.0, "GPIO Interface," on page 329.

APPLICATION NOTE: The pin control registers for GPIOs that are associated with the SDAT pins for ports supporting wake events should be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

25.4 DMA

Each EC SMB Controller can utilize two DMA Controller channels as defined in Ref [1]. DMA Channel configuration is defined in Table 24-3, “DMA Device Selection,” on page 363.

25.5 Registers

Each SMBus controller core instance in the MEC1609/MEC1609i SMB Device Interface has unique Register Interface Addressing, defined by a base address as indicated in Table 25-2. For more detail about SMBus controller core registers, see *Chapter 5, “Registers Interface”* in Ref [1].

TABLE 25-2: SMB Device Interface BASE ADDRESS TABLE

SMB Device Interface **Instance**	**LDN from (Table 3-4 on page 49)**	**AHB Base Address**
SMBus**.0**	6h	F0_1800h
SMBus**.1**		F0_1880h = F0_1800h + 80h
SMBus**.2**		F0_1900h = F0_1800h + 100h

25.6 Physical Interface

25.6.1 OVERVIEW

The Physical Interface for the SMBus 0 controller core includes eight ports as defined below in Section 25.6.2; SMBus 1 includes three ports as defined below in Section 25.6.3; SMBus 2 includes three ports and the SB-TSI port as defined below in Section 25.6.4. The port signal-function names and pin numbers are defined in Section 2.4.13, "SMBus Interface," on page 19." The SMB Device Interface port selection is made using the *PORT SEL [3:0]* bits in the *Configuration Register* as described in Ref [1] and in the subsections that follow.

For SMB Device Interface port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as defined in Section 25.6.2 and Section 25.6.3, below. For more information regarding the SMBus controller core Physical Interface, see *Section 2.2,* “*Physical Interface*” in Ref [1].

25.6.2 SMBUS 0

SMBus 0 Port Selection is defined below in Table 25-3.

TABLE 25-3: SMBus 0 PORT SELECTION

Port SEL [3:0]				**Port (Note 25-3)**
3	**2**	**1**	**0**	
0	0	0	0	SMB00
0	0	0	1	SMB01
0	0	1	0	SMB02
0	0	1	1	SMB03
0	1	0	0	SMB04
0	1	0	1	SMB05
0	1	1	0	SMB06
0	1	1	1	SMB07
1000b - 1111b				Reserved

Note 25-3 see SMBus Interface on page 19 for the SMB Device Interface pin configuration.

25.6.3 SMBUS 1

SMBus 1 Port Selection is defined below in Table 25-4.

TABLE 25-4: SMBus 1 PORT SELECTION

Port SEL [3:0]				Port (Note 25-3)
3	**2**	**1**	**0**	
0	0	0	0	SMB10
0	0	0	1	SMB11
0	0	1	0	SMB12
0011b - 1111b				Reserved

25.6.4 SMBUS 2

SMBus 2 Port Selection is defined below in Table 25-5.

TABLE 25-5: SMBus 2 PORT SELECTION

Port SEL [3:0]				Port (Note 25-3)
3	**2**	**1**	**0**	
0	0	0	0	SMB20
0	0	0	1	SMB21
0	0	1	0	SMB22
0	0	1	1	SB-TSI
0100b - 1111b				Reserved

26.0 PECI INTERFACE

26.1 Overview

The MEC1609/MEC1609i includes a PECI Interface to allow the EC to retrieve temperature readings from PECI-compliant devices. The PECI Interface implements the PHY and Link Layer of a PECI host controller as defined in References[1] and includes hardware support for PECI_REQUEST# functionality and the PECI 2.0 command set.

This chapter focuses on MEC1609/MEC1609i specific PECI Interface configuration information such as Register Addressing, Power Domain, Resets, Physical Interface, Interrupts and Clocking. For a functional description of the MEC1609/MEC1609i PECI Interface refer to References [1].

26.2 References

1. PECI Interface Core, Rev. 1.1, Core-Level Architecture Specification, SMSC Confidential

26.3 Register Addressing

The PECI Interface module is attached to EC SPB. It is assigned EC LDN 19h with base address F0_6400h; register addresses are aligned on 4-byte boundaries. The PECI Interface registers are summarized in Table 26-1. For register details see References [1].

TABLE 26-1: PECI Interface REGISTERS SUMMARY

Address Offset	Mnemonic	Register Description	NL Access (Note 26-1)
0x00	SSTWRBUF	Write Data Register	RW
0x04	SSTRDBUF	Read Data Register	RW
0x08	SSTSCTL	Control Register	RW
0x0C	SSTSTA1	Status Register 1	RWC
0x10	SSTSTA2	Status Register 2	RWC
0x14	SSTERR	Error Register	RWC
0x18	SSTINTEN1	Interrupt Enable 1 Register	RW
0x1C	SSTINTEN2	Interrupt Enable 2 Register	RW
0x20	SSTOBT1	Optimal Bit Time Register (Low Byte)	RW
0x24	SSTOBT2	Optimal Bit Time Register (High Byte)	RW
0x28	SSTRTR1	Request Timer Register (Low Byte)	RW
0x2C	SSTRTR2	Request Timer Register (High Byte)	RW
0x30-0x3C	–	Reserved	R
0x40	SSTBLKID	Block ID Register	R
0x44	SSTREV	Revision Register	R
0x48 - 0x7C	Reserved and Test Registers	MCHP Reserved. MCHP Reserved registers are reserved for use by Microchip, only. Reading and Writing MCHP Reserved registers may cause undesirable results.	RW

Note 26-1 “R” means the register is read-only, writes have no affect; “RW” means the register can written and read; “RWC” means the register can written and read but that a ‘1’ must be written to a register bit to clear it.

26.4 Block Interface Parameters

26.4.1 SIGNAL LIST

TABLE 26-2: PECI Interface Signal List

Signal Name	Type	Description
VREF_PECI	INPUT	PECI Voltage Reference pin
PECI_READY	INPUT	PECI Ready input pin (VREF_PECI). Note: PECI_READY must be pulled-up externally to enable transactions on the PECI_DAT pin.
PECI_DAT	INPUT/OUTPUT	PECI Data signal pin (VREF_PECI)
PECI_REQUEST#	OUTPUT	PECI Request output pin (VTR)
EC SPB	I/O Bus	EC MEC1609/MEC1609i peripheral bus
MCLK	INPUT	Master Clock
MCLK_DIV2_EN	INPUT	MEC1609/MEC1609i clock enable signal for PECI baud clock (32 MHz)
SPB_CLK_EN	INPUT	MEC1609/MEC1609i clock enable signal for Host interface clock
nSYS_RST	INPUT	Synchronous block reset signal
PECI_INT	OUTPUT	Interrupt signal from PECI controller to EC
SLEEP_EN	INPUT	External sleep enable control
CLOCK_REQ	OUTPUT	Clock required status
VTR	POWER	Digital logic voltage supply
GND		Ground

26.4.2 POWER DOMAIN

The PECI Interface core logic is powered by VTR; the Physical Interface Power Domain is VREF_PECI.

26.4.3 RESETS

The PECI Interface is reset on a nSYS_RST. The PECI Interface core also includes soft reset capabilities which reset control logic and part of registers. See References [1] for details.

26.4.4 CLOCKING

The PECI Interface Clocking requirement is defined below in Table 26-3.

TABLE 26-3: Clocking

Domain	Clock	Type	Frequency
CORE	MCLK	Fixed Clock	64.5 MHz
	MCLK_DIV2_EN	Fixed Clock Enable	32.25 MHz (Note 26-2)
HOST	MCLK	Fixed Clock	64.5 MHz
	SPB_CLK_EN	Variable Clock Enable	32.25 MHz and slower.

Note 26-2 Use this value for equations in References [1] that refer to the PECI Module Input Clock and affect PECI bus timing; e.g., for calculations involving the OBT register.

26.4.5 INTERRUPTS

The interrupt from the PECI Interface module is routed to the PECI_INT bit of GIRQ16 Source Register.

26.4.6 PHYSICAL INTERFACE

The pin configuration for the MEC1609/MEC1609i PECI Physical Interface is defined in Section 2.4.15, "PECI Interface," on page 19.

26.4.7 SLEEP ENABLE/CLOCK REQUIRED POWER STATE CONTROLS

For a description of the PECI Interface Sleep Enable/Clock Required Power State Controls see the PECI Interface Core, Rev. 1.1, Core-Level Architecture Specification, SMSC Confidential

27.0 ANALOG TO DIGITAL CONVERTER (ADC)

27.1 General Description

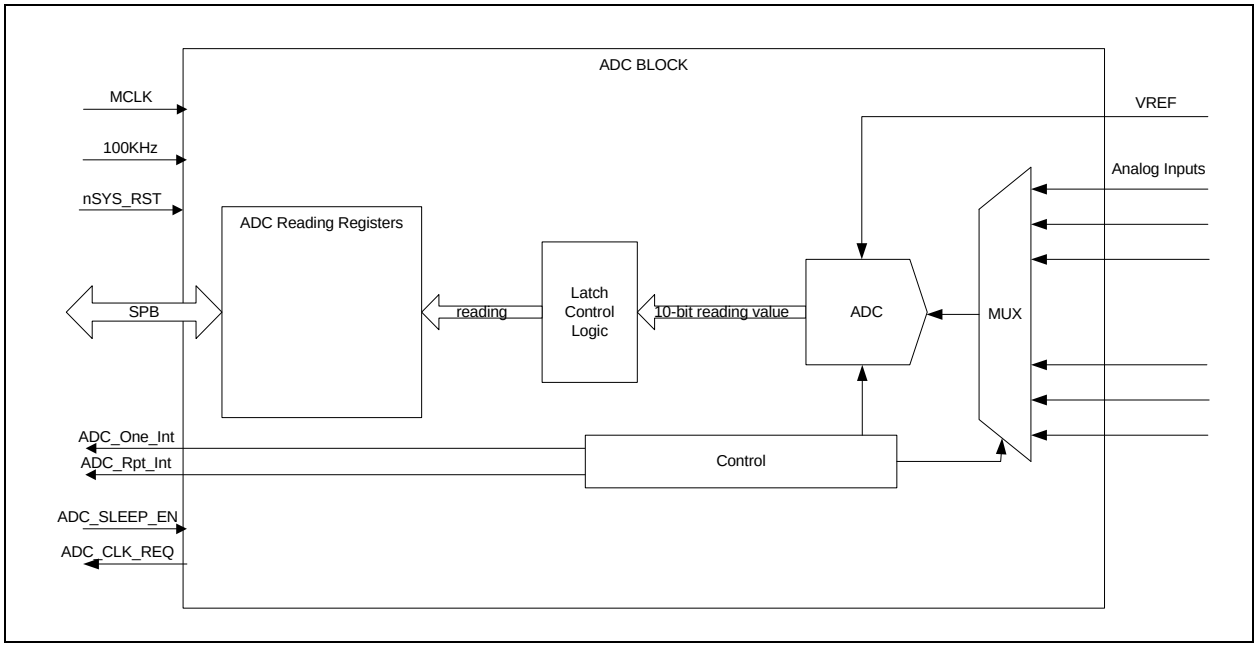
This block is designed to convert external analog voltage readings into digital values. It consists of a single successive-approximation Analog-Digital Converter that can be shared among sixteen inputs. The characteristics of this interface are shown in Table 27-1.

TABLE 27-1: ADC CHARACTERISTICS

Parameter	MIN	TYP	MAX	Unit
Resolution	–	–	10	Bits
Total Inputs (Programmable)	–	–	16	Channel
Conversion Time	–	–	10	µs/channel
Absolute Accuracy	–	2	4	LSB
Integral Non-Linearity	-0.5	–	+0.5	LSB
Differential Non-Linearity	-0.5	–	+0.5	LSB
Input Impedance	7	10	–	MOhms
Analog Input Range	0	–	VREF_ADC	Volts
VREF_ADC	2.97	–	AVTR_ADC	Volts
VREF_ADC Impedance	14 K	16 K	–	Ohms
AVTR_ADC	2.97	3.3	3.63	Volts

27.2 ADC Block Diagram

FIGURE 27-1: ADC BLOCK DIAGRAM



27.3 Block Diagram Signal List

TABLE 27-2: Analog to Digital Converter (ADC) SIGNAL LIST

Signal Name	Direction	Description
VREF_ADC	INPUT	Analog Voltage Reference
AVTR_ADC	POWER	Analog Supply
VSS_ADC	POWER	Analog Ground
SPB	I/O Bus	EC MEC1609/MEC1609i peripheral bus
MCLK	INPUT	Master MEC1609/MEC1609i clock
100KHz	INPUT	Clock enable derived from MCLK
nSYS_RST	INPUT	Block reset signal
ADC_One_Int	OUTPUT	Interrupt signal from ADC controller to EC for One-shot ADC conversion
ADC_Rpt_Int	OUTPUT	Interrupt signal from ADC controller to EC for Repeated ADC conversion
Analog Inputs	INPUT	16 analog voltage inputs from pins
ADC_SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power States on page 379.
ADC_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= 64MHz can be turned 'off' when appropriate 1= 64MHz is required to be 'on.'

27.4 Power, Clocks and Reset

27.4.1 POWER DOMAIN

This block is powered by the VTR power supply with a separate Analog supply (AVDD).

27.4.2 CLOCKS

This block has two clock inputs: MCLK at 64.52 MHz, and the 100KHz clock enable (MCLK_DIV640_EN). The latter signal is used to derive the 10ms period used for delay generation in the block. The block internally generates a 1.2MHz clock with a 50% duty cycle from the 64.52MHz. Master clock.

27.4.3 RESET

This block is reset on a nSYS_RST.

27.5 ADC Interrupts

The ADC generates an interrupt/wake-up events to the EC which indicate an ADC conversion cycle is complete. The ADC_OneStat bit and the ADC_RptStat bit in the ADC Control Register are set when conversion cycles complete. The two status bits are routed to the ADC bits in the GIRQ16 Source Register.

27.6 Low Power States

The Analog to Digital Converter (ADC) is designed to conserve power when it is either sleeping or disabled. The ADC is disabled via the Activate Bit and sleeps when the ADC_SLEEP_EN signal is asserted. The sleeping state only controls clocking in the ADC and does not power down the analog circuitry. For lowest power consumption, the ADC Activate bit must be set to ‘0.’ The following table summarizes the ADC behavior for each of these Low Power States.

TABLE 27-3: BLOCK CLOCK GATING IN LOW POWER STATES

Activate Bit	ADC_SLEEP_EN	Block Idle Status	ADC_CLK_REQ	State	Description
0	X	NOT IDLE	1	PREPARING to DISABLE	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	DISABLED	The block is idle and the core clock can be stopped.
1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

27.7 Operation

The MEC1609/MEC1609i features a sixteen channel successive approximation Analog to Digital Converter. The ADC architecture features excellent linearity and converts analog signals to 10 bit words. Conversion takes 10 microseconds per 10-bit word. The sixteen channels are implemented with a single high speed ADC fed by a sixteen input analog multiplexor. The multiplexor cycles through the sixteen voltage channels, starting with the lowest-numbered channel and proceeding to the highest-number channel, selecting only those channels that are programmed to be active.

The input range on the voltage channels spans from 0V to the external voltage reference. With a voltage reference of 3.3V, this provides resolutions of 3.2mV. The range can easily be extended with the aid of resistor dividers. The accuracy of any voltage reading depends on the accuracy and stability of the voltage reference input.

The ADC conversion cycle starts either when the Start_Once bit in the ADC Control Register to set to 1 or when the ADC Repeat Timer counts down to 0. When the Start_Once is set to 1 the conversion cycle converts channels enabled by configuration bits in the ADC One Shot Register. When the Repeat Timer counts down to 0 the conversion cycle converts channels enabled by configuration bits in the ADC Repeat Register. When both the Start_Once bit and the Repeat Timer request conversions the Start_Once conversion is completed first.

Note: If software repeatedly sets Start_Once to 1 at a rate faster than the Repeat Timer count down interval, the conversion cycle defined by the ADC Repeat Register will not be executed.

27.8 ADC Registers

The base address for the ADC block in the AHB address space is listed in Table 27-4, "Analog to Digital Converter (ADC) Base Address Table".

TABLE 27-4: Analog to Digital Converter (ADC) BASE ADDRESS TABLE

ADC **Instance**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
ADC	1Ah	F0_6800h

The following table summarizes the registers allocated for the ADC. The offset field in the following table is the offset from the Embedded Controller (EC) AHB Base Address.

TABLE 27-5: Analog to Digital Converter (ADC) REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
ADC Control Register	0h	3-0	R/W	
ADC Delay Register	4h	3-0	R/W	
ADC Status Register	8h	3-0	R/W	
ADC One Shot Register	Ch	3-0	R/W	
ADC Repeat Register	10h	3-0	R/W	
ADC Channel 0 Reading Registers	14h	3-0	R	Table 27-11
ADC Channel 1 Reading Register	18h	3-0	R	Table 27-11
ADC Channel 2 Reading Register	1Ch	3-0	R	Table 27-11
ADC Channel 3 Reading Register	20h	3-0	R	Table 27-11
ADC Channel 4 Reading Register	24h	3-0	R	Table 27-11
ADC Channel 5 Reading Register	28h	3-0	R	Table 27-11
ADC Channel 6 Reading Register	2Ch	3-0	R	Table 27-11
ADC Channel 7 Reading Register	30h	3-0	R	Table 27-11
ADC Channel 8 Reading Register	34h	3-0	R	Table 27-11
ADC Channel 9 Reading Register	38h	3-0	R	Table 27-11
ADC Channel 10 Reading Register	3Ch	3-0	R	Table 27-11
ADC Channel 11 Reading Register	40h	3-0	R	Table 27-11
ADC Channel 12 Reading Register	44h	3-0	R	Table 27-11
ADC Channel 13 Reading Register	48h	3-0	R	Table 27-11
ADC Channel 14 Reading Register	4Ch	3-0	R	Table 27-11
ADC Channel 15 Reading Register	50h	3-0	R	Table 27-11

27.8.1 ADC CONTROL REGISTER

The ADC Control Register is used to control the behavior of the Analog to Digital Converter.

TABLE 27-6: ADC CONTROL REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	00h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3-1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R	R	R	R/W	R/W	R/W
BIT NAME	ADC_ OneStat	ADC_ RptStat.	Reserved			Start_ Repeat	Start_ Once	Activate

ACTIVATE

0: The ADC is disabled and placed in a low power state. Any conversion cycle in process will complete before the block is shut down, so that the reading registers will contain valid data but no new conversion cycles will begin.

1: Start_Once or Start_Repeat can begin data conversions by the ADC. A one cycle reset pulse is sent to the ADC core when this bit changes from 0 to 1.

START_ONCE

Writing this bit with a 1 will start a single conversion cycle of all ADC channels enabled by bits Single_En[15:0] in the ADC One Shot Register. The conversion cycle will begin without a delay. Every channel that is enabled will be converted in 10 µs. After all channels enabled by Single_En[15:0] are complete, ADC_OneStat will be set to 1. When the conversion cycle begins the bit is cleared.

If Start_Once is written with a 1 while a conversion cycle is in progress because Start_Repeat is set, the conversion cycle will complete, followed immediately by a conversion cycle using Single_En[15:0] to control the channel conversions.

Writing this bit with a 0 has no effect.

START_REPEAT

Writing this bit with a 1 will start a conversion cycle of all ADC channels enabled by bits Rpt_En[15:0] in the ADC Repeat Register. The conversion cycle will begin after a delay determined by Start_Delay[15:0] in the ADC Delay Register. The Start_Delay[15:0] value is loaded into an internal Repeat Timer register and the conversion cycle begins when the Repeat Timer counts down to 0. Every channel that is enabled will be converted in 10 µs. After all channels enabled by Rpt_En[15:0] are complete, ADC_RptStat will be set to 1. As long as Start_Repeat is 1 when the Repeat Timer counts down to 0, the Repeat Timer will be reloaded with Repeat_Delay[15:0], so that the ADC will repeatedly begin conversion cycles with a period defined by Repeat_Delay[15:0]. If the delay period expires and a conversion cycle is already in progress because Start_Once was written with a 1, the cycle in progress will complete, followed immediately by a conversion cycle using Rpt_En[15:0] to control the channel conversions.

Setting this bit to 0 will not terminate any conversion cycle in process, but will clear the Repeat Timer and inhibit any further periodic conversions.

ADC_ONESTAT

This bit is cleared whenever an ADC conversion cycle is begun when Start_Once is written with a 1 and is set to 1 when the conversion cycle started by writing Start_Once completes.

This bit is also cleared when it is written with a 1. Writing a 0 to this bit has no effect.

This bit can be used to generate an EC interrupt.

ADC_RPTSTAT

This bit is cleared whenever an ADC conversion cycle is begun when Start_Repeat is 1 and is set to 1 when a repeating conversion cycle completes.

This bit is also cleared when it is written with a 1. Writing a 0 to this bit has no effect.

This bit can be used to generate an EC interrupt.

27.8.2 ADC DELAY REGISTER

The ADC Delay register determines the delay from setting Start_Repeat in the ADC Control Register and the start of a conversion cycle. This register also controls the interval between conversion cycles in repeat mode.

TABLE 27-7: ADC DELAY REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	04h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Repeat_Delay[15:8]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Repeat_Delay[7:0]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Start_Delay[15:9]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Start_Delay[7:0]							

START_DELAY[15:0]

This field determines the starting delay before a conversion cycle is begun when Start_Repeat is written with a 1. The delay is in units of 40µs. A value of 0 means no delay before the start of a conversion cycle, and a value of 0xFF means a delay of 2.6 seconds.

This field has no effect when Start_Once is written with a 1.

REPEAT_DELAY[15:0]

This field determines the interval after one conversion cycle completes and the next cycle begins when Start_Repeat is 1. The delay is in units of 40µs. A value of 0 means no delay between conversion cycles, and a value of 0xFF means a delay of 2.6 seconds.

This field has no effect when Start_Once is written with a 1.

Note:	If the Repeat Timer counts down to 0 more than once while a conversion cycle is in progress, only one periodic conversion cycle will be requested.

27.8.3 ADC STATUS REGISTER

The ADC Status Register indicates whether the ADC has completed a conversion cycle.

TABLE 27-8: ADC STATUS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	08h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**…**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	ADC_Ch_Status15	ADC_Ch_Status14	ADC_Ch_Status13	ADC_Ch_Status12	ADC_Ch_Status11	ADC_Ch_Status10	ADC_Ch_Status9	ADC_Ch_Status8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	ADC_Ch_Status7	ADC_Ch_Status6	ADC_Ch_Status5	ADC_Ch_Status4	ADC_Ch_Status3	ADC_Ch_Status2	ADC_Ch_Status1	ADC_Ch_Status0

ADC_CH_STATUS[15:0]

Each bit in this field reports the conversion status of the corresponding ADC channel. All bits are cleared either by being written with a '1,' or following a system reset (nSYS_RST). Each bit is set when the conversion on the corresponding channel is complete. When ADC_CH_Status[15:0] matches Single_En[15:0] after a conversion cycle initiated by a write to the Start_Once bit in the ADC Control Register), bit ADC_OneStat in the ADC Control Register is set and an interrupt to the EC will occur (if the interrupt is enabled). When ADC_CH_Status[15:0] matches Rpt_En[15:0] after a conversion cycle initiated by a value of 1 in bit Start_Repeat in the ADC Control Register), bit ADC_RptStat in the ADC Control Register is set and an interrupt to the EC will occur (if the interrupt is enabled).

Conversions always start with the lowest-numbered enabled channel and proceed to the highest-numbered enabled channel.

27.8.4 ADC ONE SHOT REGISTER

The ADC One Shot Register is used to control which ADC channels are captured during a one-shot conversion cycle initiated by the Start_Once bit in the ADC Control Register.

TABLE 27-9: ADC ONE SHOT REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC ADDRESS	0Ch				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Single_ En15	Single_ En14	Single_ En13	Single_ En12	Single_ En11	Single_ En10	Single_ En9	Single_ En8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Single_ En7	Single_ En6	Single_ En5	Single_ En4	Single_ En3	Single_ En2	Single_ En1	Single_ En0

SINGLE_EN[15:0]

Each bit in this field enables the corresponding ADC channel when a single cycle of conversions is started when the Start_Once bit in the ADC Control Register is written with a 1. If a Single_En[*i*] bit is 1, the channel is enabled. If a Single_En[*i*] bit is 0, the channel is disabled. At least one channel must be enabled before a conversion cycle can be initiated. Conversions start with the lowest-numbered channel that is enabled and proceed to the highest-numbered enabled channel. If this register is changed while a conversion cycle is in progress the conversion cycle will use the new values for channels that have not yet been examined, but will not rescan channels that have already been checked.

27.8.5 ADC REPEAT REGISTER

The ADC Repeat Register is used to control which ADC channels are captured during a one-shot conversion cycle initiated by the Start_Repeat bit in the ADC Control Register.

TABLE 27-10: ADC REPEAT REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	10h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Rpt_ En15	Rpt_ En14	Rpt_ En13	Rpt_ En12	Rpt_ En11	Rpt_ En10	Rpt_ En9	Rpt_ En8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Rpt_ En7	Rpt_ En6	Rpt_ En5	Rpt_ En4	Rpt_ En3	Rpt_ En2	Rpt_ En1	Rpt_ En0

RPT_EN[15:0]

Each bit in this field enables the corresponding ADC channel for each pass of the Repeated ADC Conversion that is controlled by bit Start_Repeat in the ADC Control Register. If a Rpt_En[*i*] bit is 1, the channel is enabled. If a Rpt_En[*i*] bit is 0, the channel is disabled. At least one channel must be enabled before a conversion cycle can be initiated. Conversions start with the lowest-numbered channel that is enabled and proceed to the highest-numbered enabled channel. If this register is changed while a conversion cycle is in progress the conversion cycle will use the new values for channels that have not yet been examined, but will not rescan channels that have already been checked.

27.8.6 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 27-11 shows the format of all the reading registers. Table 27-5, “Analog to Digital Converter (ADC) Register Summary,” on page 381 shows the addresses of all the reading registers.

TABLE 27-11: ADC CHANNEL X READING REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	xxh			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**…**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved						ADCx_[9:8]	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ADCx_[7:0]							

ADCX_[9:0]

This read-only field reports the 10-bit output reading of ADCx.

28.0 TACH MONITOR

28.1 General Description

This block is designed to monitor tach output signals or locked rotor signals from various types of fans to determine their speed. One mode returns the value in number of CLOCK_LOW pulses. Another mode returns the value in pulses per programmed amount of time. This second mode can use the raw tach input. Each Tach is associated with a pair of limit registers that define maximum and minimum acceptable Tach counter values. If the readings on a Tach is outside these limits an interrupt to the EC can be generated.

In typical systems the fans are powered by the main power supply. Firmware may disable this block when it detects the main power rail has been turned off.

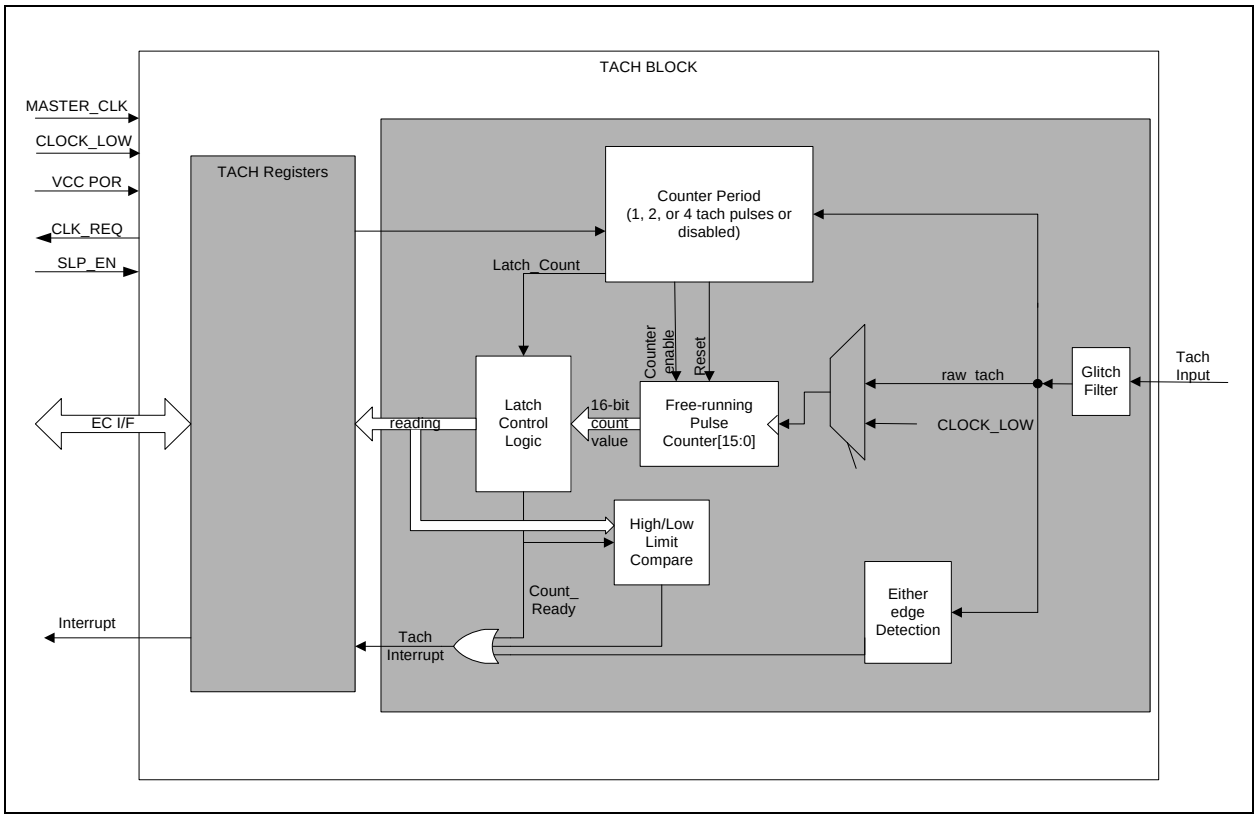
APPLICATION NOTE: This block can be utilized with Fans running at the following speed range:100 to 30K RPM.

The TACH Monitor performs the following functions:

- Count the number of pulses detected on the raw tach input.
- Count the number of clocks for a programmed number of pulses.
- Generate an interrupt when the count value is latched into the reading register.
- Generate a programmable either-edge triggered interrupt for detecting when the tach input changes state. This may be used for Locked Rotor detection.
- Generate an interrupt when the count value latched into the reading register is greater than the high limit or less than the low limit.

28.2 TACH Monitor Block Diagram

FIGURE 28-1: BLOCK DIAGRAM OF TACH MONITOR



Note:	Once the counter is enabled it is a 16-bit free-running counter. Latch count value on a read or when number of tach pulses is detected (if enabled) for 1, 2, or 4 pulses. Counter is reset to 0000h if the count value is latched by a programmed number of tach pulses and on a VCC POR. Counter enable is software controlled signal.

28.3 Block Diagram Signal List

TABLE 28-1: TACH PORT LIST

Signal Name	Direction	Description
VTR POR	INPUT	nSYS_RST
Master_Clock	INPUT	64.52MHz MCLK
CLOCK_LOW	INPUT	100KHz MCLK_DIV640_EN
EC I/F	I/O Bus	EC-side SPB bus
Tach Input	INPUT	Tachometer signal from TACHx Pin
Interrupts	OUTPUT	Interrupt used to indicate that either Tach Input has changed state or the TACH reading has been updated. One per TACH
SLP_EN	INPUT	Sleep Enable input from MEC1609/MEC1609i Clock Generator Power Management Interface.
CLK_REQ	OUTPUT	Clock Required output to MEC1609/MEC1609i Clock Generator Power Management Interface.

28.4 Power, Clocks and Reset

28.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

28.4.2 CLOCKS

This block uses the EC Bus Clock and the 100KHz MCLK_DIV640_EN. EC Bus Clock is used when reading and writing the TACH Monitor control registers. The individual TACH counters are driven by Clock_Low, the MCLK_DIV640_EN.

The TACH Monitor clock required output (CLK_REQ) is the inversion of the sleep enable input (SLP_EN). The CLK_REQ output is not asserted when the TACH Monitor is disabled.

See also Section 5.1.2, "Clock Generator," on page 73 for details on clocks.

28.4.2.1 Clock Idle

When the internal ring oscillator is disabled or when the TACH block is disabled, the internal TACH counters are reset. The reading register is not affected. This insures that inaccurate readings are not generated if the master clock halts in the middle of a TACH reading or when the TACH starts up.

Note:	Each Tach pin should be pulled up via an external resistor to the main power supply.

28.4.3 RESET

This block is reset on a nSYS_RST.

See Section 5.1.3, "Reset Interface," on page 73 for details on reset.

28.5 TACH Interrupts

Each TACH Monitor in the MEC1609/MEC1609i can be used to generate one interrupt event. Each TACH Monitor interrupt source is a level, active high signal. The TACH Monitor interrupts are routed to the TACH3, TACH2, TACH1, & TACH0 bits in GIRQ17 Source Register on page 278. The TACH Monitor interrupts generate interrupt events.

28.6 TACH Circuitry

The TACH Circuitry is implemented as a pulse counter. There are two types of toggling signals that can be used to increment the counter: the raw tach input or CLOCK_LOW. See FIGURE 28-1: Block Diagram of TACH Monitor on page 388. The two modes for incrementing the counter are controlled by Tach Reading Mode Select in the TACHx Control Register.

If the raw tach is used to increment the counter, the circuitry can be configured as a free-running counter that increments when a pulse from the tach is detected (i.e., input signal transitions from low-to-high).The counter is latched into the reading register (Tachx Counter in the TACHx Control Register) every time it is incremented. If this mode is selected, firmware will monitor the number of pulses detected over a period of time to determine the speed of the attached fan.

If CLOCK_LOW is used to increment the counter, the raw tach input will be used to determine when to latch the current count value into the reading register and reset the counter to 0000h. The counter is latched after a programmed number of tach pulses is detected. The programmed period can be configured to be 1, 2, or 4 tach pulses in duration.

Each Tach counter has comparison logic to compare the counter value with the high limit and low limit registers.

28.6.1 TACH INTERRUPT SOURCES

There are three interrupt source events: notify EC when reading is updated, notify EC when TACH input toggles, or notify EC when the TACH reading exceeds a programmed limit. The corresponding interrupt status bits are Count Ready Status Toggle Status Out-of-Limit Status Bits[3,1,0] in TACHx Status Register on page 394.

28.6.1.1 Count Reading Ready Status

This status bit is asserted when the counter value is latched. The bit is implemented in Bit D3 of the TACHx Status Register.

28.6.1.2 Tach Input Toggle Status

This status bit is asserted when the Tach input changes state. The bit is implemented in Bit D2 of the TACHx Status Register.

28.6.1.3 TACH Out-of_Limit STATUS

To generate a TACH out-of-limit status event, the high and low limits may be programmed in the TACHx High Limit Register and TACHx Low Limit Register. An out-of-limit event is triggered when the reading register (Tachx Counter in the TACHx Control Register) is set to a value less than the TACHx Low Limit Register or to a value greater than the TACHx High Limit Register. If the value in the Tachx Counter violates the programmed limits the TACH limit registers a status event will be generated, indicating the out-of-limit event. This status bit is implemented in Bit D0 of the TACHx Status Register. This signal may be used to interrupt the Embedded Controller, if enabled via Bit D0 of the TACHx Control Register.

Note:	If the TACHx Low Limit Register is set to 0000h, no out-of-limit event will be triggered by a Tachx Counter value that is below the limit. If the TACHx High Limit Register is set to FFFFh, no out-of-limit event will be triggered by a Tachx Counter value that is above the limit.

APPLICATION NOTE: Out-of-Limit checks are typically only used when the tach counter is incremented in Mode 1 (in which the counter counts the number of CLOCK_LOW until a programmed number of pulses occur on the raw tach input).

28.7 Registers

There are four block instances defined in this chapter: TACH[3:0].

Each instance of the TACH Monitor has its own Logical Device Number, and Base Address as indicated in Table 28-2.

TABLE 28-2: TACH Monitor BASE ADDRESS TABLE

TACH Monitor Instances	LDN from (Table 3-4 on page 49)	AHB Base Address
TACH**0**	18h	F0_6000h
TACH**1**		F0_6080h
TACH**2**		F0_6100h
TACH**3**		F0_6180h

Table 28-3 is a register summary for one instance of the TACH Monitor.

TABLE 28-3: TACHX REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
TACHx Control Register	00h	3-0	R/W	
TACHx Status Register	04h	0	R/W	
TACHx High Limit Register	08h	1-0	R/W	
TACHx Low Limit Register	0Ch	1-0	R/W	

28.7.1 DETAILED DESCRIPTION OF TACHOMETER REGISTER VALUES

This section describes the parameters that must be stored in hardware registers that will used by the TACH logic.

28.7.1.1 TACHx Control Register

TABLE 28-4: TACHX CONTROL REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	00h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	TACHx Counter [15:8] Register							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	TACHx Counter [7:0] Register							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Tach Input INT_EN	Count Ready INT_EN	Reserved	Tach Edges		Tach Reading Mode Select	Reserved	Filter Enable
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						TACH Enable	Tach Out-of-Limit Enable

TACH OUT-OF-LIMIT ENABLE

The TACH Out-of_Limit Enable is used to enable Bit[0] TACH Out-of_Limit Status bit to generate an interrupt event.

0=disable interrupt output from tach block (default)

1=enable interrupt output from tach block

TACH ENABLE

This bit enables the TACH logic.

0=TACH Idle (default)

This mode gates the clocks to the TACH block. The TACHx pin is tristate in the idle mode.

1=TACH Monitoring enabled

APPLICATION NOTE: This bit gates the clocks into the block. When re-enabled, the internal counters will continue from the last known state and stale status events may still be pending. Firmware should

discard any status or reading values until the reading value has been updated at least one time after the enable bit is set.

FILTER ENABLE

The TACH glitch filter rejects input pulses that are less than three CLOCK_LOW periods wide.

0=Filter disabled (default)

1=Filter enabled

TACH READING MODE SELECT

0 = Counter is incremented when Tach Input transitions from low-to-high state (default)

1 = Counter is incremented on the rising edge of the CLOCK_LOW input. The counter is latched into Tachx Counter and reset when the programmed number of edges is detected.

TACH EDGES

A tach signal is a square wave with a 50% duty cycle. Typically, two tach periods represents one revolution of the fan. A tach period consists of three tach edges.

This programmed value represents the number of tach edges that will be used to determine the interval for which the number of CLOCK_LOW pulses will be counted.

00 = 2 Tach edges (1/2 tach period)

01 = 3 Tach edges (1 tach period)

10 = 5 Tach edges (2 tach periods)

01 = 9 Tach edges (4 tach periods)

COUNT READY INT_EN

0=disable interrupt output from tach block (default)

1=enable interrupt output from tach block

TACH INPUT INT_EN

0=disable interrupt output from tach block (default)

1=enable interrupt output from tach block

TACHX COUNTER

This 16-bit field contains the latched value of the tach counter, which may be configured to operate as a free-running counter or to be gated by the tach input signal.

If the counter is free-running (Mode 0), it increments (if enabled) at the rate determined by the raw tach signal and latched into this field every time it is incremented. The act of reading this field will not reset the counter, which rolls over to 0000h after FFFFh. The firmware will compute the delta between the current count reading and the previous count reading, to determine the number of pulses detected over a programmed period.

If the counter is being gated by the tach input and clocked by the CLOCK_LOW (Mode 1), the counter will be latched into the reading register when the programmed number of edges is detected or when the counter reaches FFFFh and the counter will be reset to zero.

APPLICATION NOTE: In Mode 1, a counter rate of FFFFh means that the tach did not detect the programmed number of edges in 655ms. A stuck fan can be detected by setting the TACHx High Limit Register to a number less than FFFFh. If the counter then reaches FFFFh, the reading register will be set to FFFFh and an out-of-limit interrupt can be sent to the EC.

28.7.1.2 TACHx Status Register

TABLE 28-5: TACHX STATUS REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3-1 BIT	**D31**	**D32**	**D31**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/WC	R/WC	R	R/WC
BIT NAME	Reserved				Count Ready Status	Toggle Status	TACH Pin Status	Tach Out-of-Limit Status

TACH OUT-OF-LIMIT STATUS

This bit is set when the Tach Count value is greater than the high limit or less than the low limit. It is cleared when written with a 1. To disable this status event set the limits to their extreme values. If enabled via TACH Out-of-Limit Enable in the TACHx Control Register, this status bit will assert the Interrupt signal, which may be enabled to cause an interrupt event to the embedded controller.

0=Within Limits (TACH count value is less than or equal to the high limit or greater than or equal to the low limit).

1=Out of Limits (TACH count value is greater than the high limit or less than the low limit).

TACH PIN STATUS

This bit reflects the state of Tach Input. This bit is a read only bit that may be polled by the embedded controller.

0=Tach Input is low

1=Tach Input is high

TOGGLE STATUS

This bit is set when Tach Input changes state. It is cleared when written with a 1. If enabled via Tach Input INT_EN in the TACHx Control Register, this status bit will assert the Interrupt signal, which may be enabled to cause an interrupt event to the embedded controller.

0=Tach stable (default)

1=Tach Input changed state (this bit is set on a low-to-high or high-to-low transition)

APPLICATION NOTE: Some fans offer a Locked Rotor output pin that generates a level event if a locked rotor is detected. This bit may be used in combination with the tach pin status bit to detect a locked rotor signal event from a fan.

APPLICATION NOTE: Tach Input may come up as active for Locked Rotor events. This would not cause an interrupt event because the pin would not toggle. Firmware must read the status events as part of the initialization process, if polling is not implemented.

COUNT READY STATUS

The Count Ready Status bit remains cleared to '0' when the Tach Reading Mode Select bit in the TACHx Control Register is clear to '0'.

When the Tach Reading Mode Select bit in the TACHx Control Register is set to '1', The Count Ready Status bit is set when the counter value is latched by the hardware. It is cleared when written with a 1. If enabled via the Count Ready INT_EN bit in the TACHx Control Register, this status bit will assert the Interrupt signal, which may be enabled to cause an interrupt event to the embedded controller.

0=Reading not ready

1=Reading ready

28.7.1.3 TACHx High Limit Register

TABLE 28-6: TACHX HIGH LIMIT REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSETS	08h				32-bit		**EC SIZE**	
POWER	VTR				FFFFh		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**…**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1-0 BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACHx High Limit [15:0] Register							

The TACHx High Limit [15:0] value is compared with the value in the TACHx Control Register. If the value in the TACHx Control Register is greater than the value programmed in the TACHx High Limit Register the TACH Out-of_Limit STATUS bit will be set. The TACH Out-of-Limit status event may be enabled to generate an interrupt to the embedded controller via Bit[0] of the TACHx Control Register.

Note: To disable this event program FFFFh into this register.

28.7.1.4 TACHx Low Limit Register

TABLE 28-7: TACHX LOW LIMIT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSETS	0Ch			32-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3-2 BIT	**D31**	**D29**	**D28**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACHx Low Limit [15:0] Register							

The TACHx Low Limit [15:0] value is compared with the value in the Tachx Counter Field of the TACHx Control Register. If the value in the Tachx Counter Field is less than the value programmed in the TACHx Low Limit Register the TACH Out-of_Limit STATUS bit will be set. The TACH Out-of-Limit status event may be enabled to generate an interrupt to the embedded controller via Bit[0] of the TACHx Control Register.

To disable this event program 0000h into this register.

29.0 PWM CONTROLLER

29.1 General Description

The function of this block is to generate a PWM output that may be used to control 4-wire fans, blink LEDs, etc. Each PWM can generate an arbitrary duty cycle output at frequencies from 0.125Hz to 32MHz.

The PWMx Counter ON Time registers and PWMx Counter OFF Time registers determine the operation of the PWM_OUTPUT signal. See Section 29.3.1, "PWMx Counter ON/OFF Time Registers," on page 401 for a description of the PWM_OUTPUT signal.

29.1.1 PWM_OUTPUT

The PWM_OUTPUT signal is used to generate a duty cycle at a frequency. This block has been designed such that the PWM signal may be programmed to hold PWM_OUTPUT high, to hold PWM_OUTPUT low, or to toggle PWM_OUTPUT. If the PWM is configured to toggle, then PWM_OUTPUT will alternate high and low for the programmed duration in the PWMx Counter ON/OFF Time Registers registers as defined in the register description. The PWM equations are described in Figure 29-2.

The PWM pin signal functions are routed pins described in Table 2-12, “Fan PWM & Tachometer Interface,” on page 15.

29.1.2 PWM FEATURES

APPLICATION NOTE: Each PWM pin signal functions is muxed with a GPIO pin signal function. The pin’s default signal function is GPIO input as controlled by the associated Pin Control Register. (See Section 22.0, "GPIO Interface," on page 329). At VTR POR or when a WDT event occurs (see Section 17.0, "Watchdog Timer Interface," on page 293,) the pin will tristate. For fan applications, an external resistor termination can provide the pin state to force the external fans to the full on state, thereby protecting the system from overheating.

29.1.3 PWM CONTROLLER BLOCK DIAGRAM

FIGURE 29-1: PWM FUNCTIONAL DIAGRAM

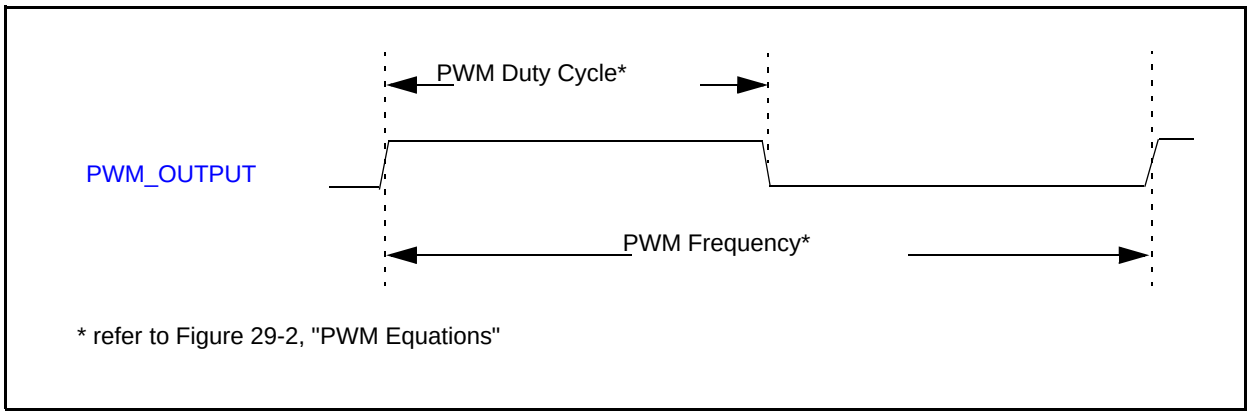


FIGURE 29-2: PWM EQUATIONS

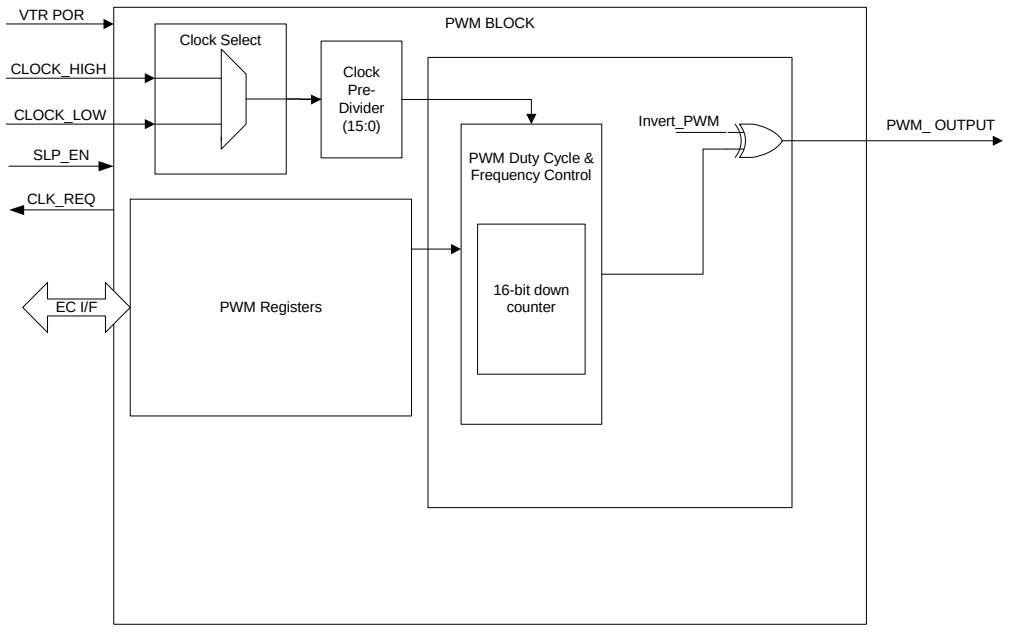
$$\textbf{PWM Frequency} = \frac{1}{(\text{PreDivider} + 1)} X \frac{(\overline{\text{Clock Select}} * 64.52\ \text{MHz} \ \&\ \text{Clock Select} * (64.52\ \text{MHz}/640))}{(\text{PWM Counter OFF Time}[15:0] + 1) + (\text{PWM Counter ON Time}[15:0] + 1)}$$

$$\textbf{PWM Duty Cycle} = \frac{(\text{PWM Counter ON Time}[15:0] + 1)}{(\text{PWM Counter OFF Time}[15:0] + 1) + (\text{PWM Counter ON Time}[15:0] + 1)}$$

Legend:
* = a Boolean AND operator
& = a Boolean OR operator

Notes:
Clock Select, PreDivider, PWM Counter OFF Time[15:0], and PWM Counter ON Time[15:0] are register values defined below.

FIGURE 29-3: BLOCK DIAGRAM OF PWM CONTROLLER



29.1.3.1 Block Diagram Signal List

TABLE 29-1: BLOCK DIAGRAM SIGNAL LIST DESCRIPTION

Signal Name	Direction	Description
VTR POR	INPUT	VTR Power on Reset.
CLOCK_HIGH	INPUT	64.52MHz MCLK.
CLOCK_LOW	INPUT	100KHz MCLK_DIV640_EN.
PWM_OUTPUT	OUTPUT	Pulse Width Modulated signal to PWMx pin.
E/C IF	I/O Bus	EC-side SPB bus.
SLP_EN	INPUT	Sleep Enable input.
CLK_REQ	OUTPUT	Clock Required output.

29.2 Power, Clocks and Reset

29.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

29.2.2 CLOCKS

This block uses the EC Bus Clock, the 64.52MHz MCLK and the 100KHz MCLK_DIV640_EN. The EC Bus Clock is used when reading and writing the PWM Controller control registers. The individual PWM counters can be driven either by MCLK or MCLK_DIV640_EN.

See Section 5.1.2, "Clock Generator," on page 73 for details on clocks.

29.2.2.1 Pre-Divider

The clock source to the PWM Down Counter used to generate a duty cycle and frequency on the PWM_OUTPUT may be pre-divided via bits D6:D3 in the PWMx Configuration Register. This results in a wide range of frequencies for the pwm output. Table 29-2 shows examples of frequencies supported.

TABLE 29-2: EXAMPLE OF PWM FREQUENCIES DERIVED USING PRE-DIVIDER

Ex. #	Clock (MHz)	Clock Select	Clock Pre-Divider (0 - 15)	High Count	Low Count	PWM_OUTPUT Frequency (Hz)	PWM_OUTPUT Duty Cycle (%)
1	64.52	0	0	32768	32768	984.467	50.0%
2	64.52	0	0	192	192	167150.259	50.0%
3	64.52	0	0	382	2	167150.259	99.2%
4	64.52	0	0	960	960	33569.199	50.0%
5	64.52	0	0	32767	32767	984.497	50.0%
6	0.1008125	1	1	32767	32767	0.769	50.0%
7	0.1008125	1	11	32767	32767	0.128	50.0%

29.2.2.2 Sleep Enable

The Embedded Controller can put each PWM into a sleep state. When a PWM is in the sleep state the internal counters are reset to 0 and the internal state of the PWM and thus the PWM_OUTPUT signal is set to the OFF state.

The PWM Controller clock required output (CLK_REQ) is the inversion of the sleep enable input (SLP_EN). The CLK_REQ output is not asserted when the PWM Controller is disabled.

The PWM participation in the sleep state is controlled by the PWMx bits in the EC Blocks Sleep Enables Register 1 and EC Blocks Sleep Enables Register 2.

29.2.3 RESET

This block is reset by nSYS_RST. After the assertion of nSYS_RST, PWM_OUTPUT is held in the OFF state and the hardware resets the pwm counter registers to their default value.

See Section 5.1.3, "Reset Interface," on page 73 for details on reset.

29.3 Registers

There are eight instances of the PWM Controller block implemented in the MEC1609/MEC1609i enumerated [7:0]. Each instance of the PWM Controller has its Base Address as indicated in Table 29-3, "PWMx Controller Base Address Table":

TABLE 29-3: PWMX CONTROLLER BASE ADDRESS TABLE

PWM Controller **Instance**	**LDN from (Table 3-4 on page 49)**	**AHB Base Address**
PWM**.0**	16h	F0_5800h
PWM**.1**		F0_5880h
PWM**.2**		F0_5900h
PWM**.3**		F0_5980h
PWM**.4**		F0_5A00h
PWM**.5**		F0_5A80h
PWM**.6**		F0_5B00h
PWM**.7**		F0_5B80h

Table 29-4 summarizes the registers allocated for each Instance. The offset field in the following table is the offset from the Embedded Controller (EC) Base Address.

TABLE 29-4: PWMX REGISTER SUMMARY

	EC Interface			**Notes**
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	
PWMx Counter ON Time Register	00h	1-0	R/W	
PWMx Counter OFF Time Register	04h	1-0	R/W	
PWMx Configuration Register	08h	1-0	R/W	

TABLE 29-5: PWMX EC ACCESSIBLE REGISTERS

Offset	**Register Name**	**VTR POR (Suspend)**
0h	PWMx Counter ON Time Register	0000h
4h	PWMx Counter OFF Time Register	FFFFh
8h	PWMx Configuration Register	0000h

29.3.1 PWMX COUNTER ON/OFF TIME REGISTERS

TABLE 29-6: PWMX COUNTER ON TIME REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	00h				16-bit		**EC SIZE**	
POWER	VTR				0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	PWMx Counter ON Time[15:0]							

TABLE 29-7: PWMX COUNTER OFF TIME REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	04h				16-bit		**EC SIZE**	
POWER	VTR				FFFFh		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	PWMx Counter OFF Time[15:0]							

The PWMx Counter ON/OFF Time registers determine both the duty cycle and frequency of the signal generated on PWM_OUTPUT. See FIGURE 29-2: PWM Equations on page 398.

If the PWMx Counter OFF Time[15:0] is set to zero, PWM_OUTPUT is held high (Full On). If the PWMx Counter ON Time is set to zero and the PWMx Counter OFF Time[15:0] is not set to zero, PWM_OUTPUT is held low (Full Off). Note that the default case is full off. Otherwise, both the high and low count registers will contain a value that will be used to determine the length of time PWM_OUTPUT will be held high and low. See Table 29-8, "PWM_OUTPUT State".

TABLE 29-8: PWM_OUTPUT STATE

PWM Count ON Time	PWM Count OFF Time	State Of PWM_OUTPUT
Don’t Care	0000h	Full On
0000h	Non-Zero Value	Full Off
Non-Zero Value	Non-Zero Value	Toggling On and Off

The counter values preload a 16-bit down-counter that is clocked by either the high frequency clock source or the low frequency clock source (see bit[1] CLK_Select of PWMx Configuration Register). The firmware will program the on and off count values that correspond to the PWM Current Duty Cycle and PWM Frequency. When PWM_OUTPUT is OFF and the internal counter is zero, the PWMx Counter ON Time is loaded into the counter. The PWM_OUTPUT signal will transition to the ON state and the internal counter will count down to zero at the programmed frequency for the duration of the programmed on time. Similarly, when the PWM_OUTPUT is in the ON state and the internal counter is zero, the PWMx Counter OFF Time is loaded into the counter. The PWM_OUTPUT signal will transition OFF and the internal counter will count down to zero at the programmed frequency for the duration of the programmed off time.

The PWMx Counter ON/OFF Time Registers may be updated at any time. Values written into the two registers are kept in holding registers. The holding registers are transferred into the PWMx Counter ON/OFF Time Registers when all four bytes have been written with new values and the internal counter completes the OFF time count. If the PWM is in the

Full On state then the PWMx Counter ON/OFF Time Registers are updated from the holding registers as soon as all four bytes have been written. Once the two registers have been updated the holding registers are marked empty. and all four bytes must again be written before the holding registers will be reloaded into the PWMx Counter ON/OFF Time Registers. Reads of the PWMx Counter ON/OFF Time Registers return the current contents of the registers that are used to load the counter and not the holding registers.

29.3.2 PWMX CONFIGURATION REGISTER

TABLE 29-9: PWMX CONFIGURATION REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	08h				16-bit		**EC SIZE**	
POWER	VTR				0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Clock Pre-Divider				Invert	Clock Select	PWM Enable

PWM ENABLE

0= disabled (gates clocks to save power) (default)

1= enabled

Note:	When the PWM enable bit is set to 0 the internal counters are reset and the internal state machine is set to the OFF state. In addition, the PWM_OUTPUT signal is set to the inactive state as determined by the Invert bit. The PWMx Counter ON/OFF Time Registers are not affected by the PWM enable bit and may be read and written while the PWM enable bit is 0.

CLOCK SELECT

The Clk_Select bit determines the clock source used by the PWM duty cycle and frequency control logic.

0= 64.52MHz MCLK (default)

1= 100KHz MCLK_DIV640_EN

INVERT

0= PWM_OUTPUT ON State is active high

1= PWM_OUTPUT ON State is active low

CLOCK PRE-DIVIDER

The Clock source for the 16-bit down counter (see PWMx Counter ON/OFF Time Registers) is determined by bit D1 of this register. The Clock source is then divided by the value of Pre-Divider+1 and the resulting signal determines the rate at which the down counter will be decremented. For example, a Pre-Divider value of 1 divides the input clock by 2 and a value of 2 divides the input clock by 3. A Pre-Divider of 0 will disable the Pre-Divider option.

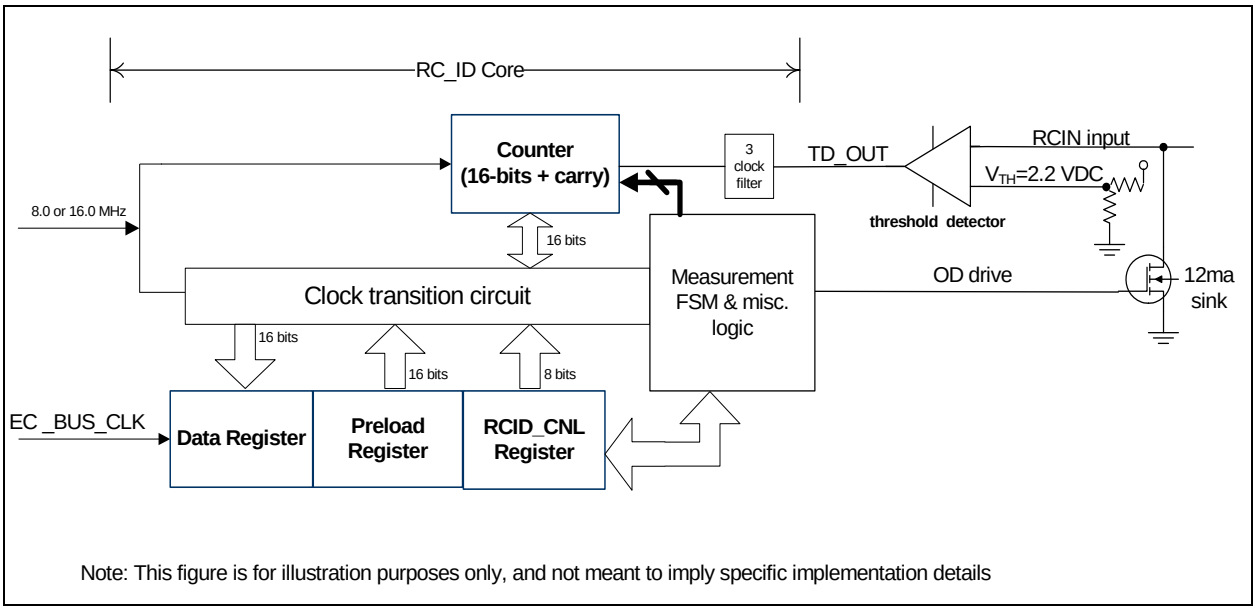
30.0 RC IDENTIFICATION DETECTION (RC_ID)

30.1 General Description

The Resistor/Capacitor Identification Detection (RC_ID) interface provides a single pin interface which can discriminate a number of quantized RC constants. The judicious selection of RC values can provide a low cost means for system element configuration identification. The RC_ID I/O pin measures the charge/discharge time for an RC circuit connected to the pin as shown in Figure 30-1.

30.2 Block Diagram

FIGURE 30-1: BLOCK DIAGRAM OF RC Identification Detection (RC_ID)



Note: This figure is for illustration purposes only, and not meant to imply specific implementation details

30.3 Power, Clocks and Reset

30.3.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

30.3.2 CLOCKS

This block uses the EC Bus Clock and the 64.52MHz MCLK. The EC Bus Clock is used to access the Registers described in this block. MCLK is divided down to provide a sampling clock.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

30.3.3 POWER ON RESET

This block is reset on a nSYS_RST. Following a system reset, all Register are reset to 00h and the state machines are set to Reset.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

30.4 Interrupts

The RC Identification Detection (RC_ID) can generate an RCID_DONE interrupt when the DONE bit in the RCID_CTL Register is set. The interrupt source is routed onto the RCID bit in GIRQ16 Source Register on page 277 and is a level, active high signal.

30.5 Time Constants

This section lists a set of R and C values which can be connected to the RC_ID pin. Note that risetime generally follows RC time Tau; however empirical characterization is required. Firmware should use the Max and Min Limits to create quantized states.

30.5.1 CHARACTERIZED SPECIFIC TIME CONSTANTS - 16 MHZ CLOCK

TABLE 30-1: SAMPLE RC VALUES (C=2200 PF. R VARIED)

LIMITS				External Circuit Components		
MIN count	**MAX count**	**Range**	**Band Gap**	**C (pF)**	**R (K)**	
Avg-20%	Avg+10%	max-min	between RC values	**10%**	**5%**	
35	50	15		2200	1	
65	91	26	15	2200	2	
134	186	52	43	2200	4.3	
252	348	96	66	2200	8.2	
994	1367	373	646	2200	33	
1850	2545	695	483	2200	62	
3778	5197	1419	1233	2200	130	
6880	9461	2581	1683	2200	240	
MIN	**MAX**	**Range**		**C (pF)**	**R (K)**	**Tau =RxC**
Risetime (usec)	**Risetime (usec)**	max-min		**10%**	**5%**	**(usec)**
2.36	4.00	1.64		2200	1	2.20
4.58	6.29	1.72		2200	2	4.40
9.41	12.94	3.53		2200	4.3	9.46
17.42	23.95	6.53		2200	8.2	18.04
64.05	88.06	24.02		2200	33	72.60
113.26	155.74	42.47		2200	62	136.40
220.15	302.70	82.55		2200	130	286.00
378.52	520.47	141.95		2200	240	528.00

TABLE 30-2: SAMPLE RC VALUES (C=3000 PF. R VARIED)

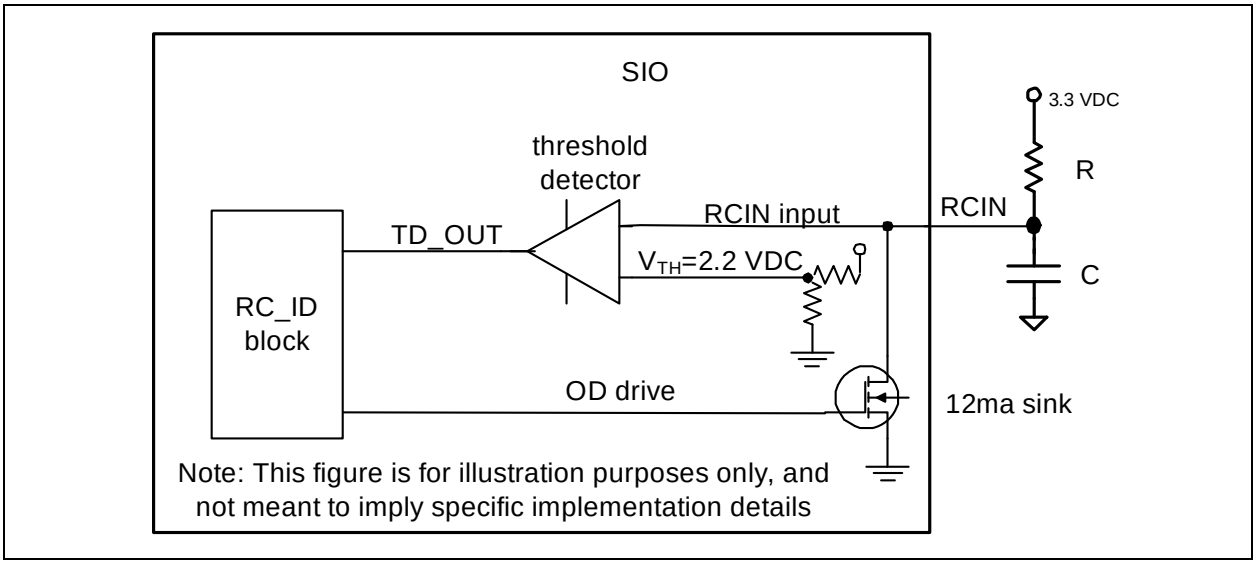
LIMITS				External Circuit Components		
MIN count	**MAX count**	**Range**	**Band Gap**	**C (pF)**	**R (K)**	
Avg-20%	Avg+10%	max-min	between RC values	**10%**	**5%**	
48	68	20		3000	1	
92	128	36	24	3000	2	
194	268	74	66	3000	4.3	
369	509	140	101	3000	8.2	
1482	2038	556	973	3000	33	
2777	3820	1043	739	3000	62	
5728	7877	2149	1908	3000	130	
10482	14414	3932	2605	3000	240	
MIN	**MAX**	**Range**		**C (pF)**	**R (K)**	**Tau =RxC**
Risetime (usec)	**Risetime (usec)**	max-min		**10%**	**5%**	**(usec)**
3.33	5.00	1.67		3000	1	3.00
6.58	9.04	2.47		3000	2	6.00
13.62	18.73	5.11		3000	4.3	12.90
25.13	34.56	9.42		3000	8.2	24.60
92.86	127.68	34.82		3000	33	99.00
166.54	228.99	62.45		3000	62	186.00
326.99	449.61	122.62		3000	130	390.00
565.18	777.12	211.94		3000	240	720.00

TABLE 30-3: SAMPLE RC VALUES (C=4700 PF. R VARIED)

LIMITS				External Circuit Components			Measured		
MIN count	**MAX count**	**Range**	**Band Gap**	**C (pF)**	**R (K)**		**MIN count**	**MAX count**	**Mean**
Avg-20%	Avg+10%	max-min	between RC values	**10%**	**5%**				
72	101	29		4700	1		90	92	91.11
141	195	54	40	4700	2		174	179	176.70
299	412	113	104	4700	4.3		369	378	374.07
567	780	213	155	4700	8.2		700	718	708.93
2255	3102	847	1475	4700	33		2770	2870	2819.19
4215	5797	1582	1113	4700	62		5148	5406	5269.22
8686	11945	3259	2889	4700	130		10460	11288	10858.70
15887	21846	5959	3942	4700	240		18786	21100	19859.41
MIN Risetime (usec)	**MAX Risetime (usec)**	**Range** max-min		**C (pF) 10%**	**R (K) 5%**	**Tau =RxC (usec)**	**MIN Risetime (usec)**	**MAX Risetime (usec)**	**Mean Risetime (usec)**
4.80	6.65	1.84		4700	1	4.70	5.86	6.29	6.01
9.62	13.40	3.77		4700	2	9.40	11.44	12.36	12.03
20.44	28.05	7.61		4700	4.3	20.21	23.84	26.20	25.55
38.90	53.67	14.77		4700	8.2	38.54	46.28	49.78	48.62
152.13	212.63	60.50		4700	33	155.10	175.30	194.60	190.16
275.34	381.48	106.14		4700	62	291.40	334.80	350.80	344.18
489.30	658.24	168.94		4700	130	611.00	593.10	639.10	611.63
677.53	949.63	272.10		4700	240	1128.00	816.00	881.40	846.92

30.6 Block Diagram

FIGURE 30-2: RCID CIRCUIT.INTERFACING TO THE RC_ID BLOCK



The RC_ID uses the 8.0 / 16.0MHz clock input; therefore, a measurement can only be made when VTR and this clock are present.

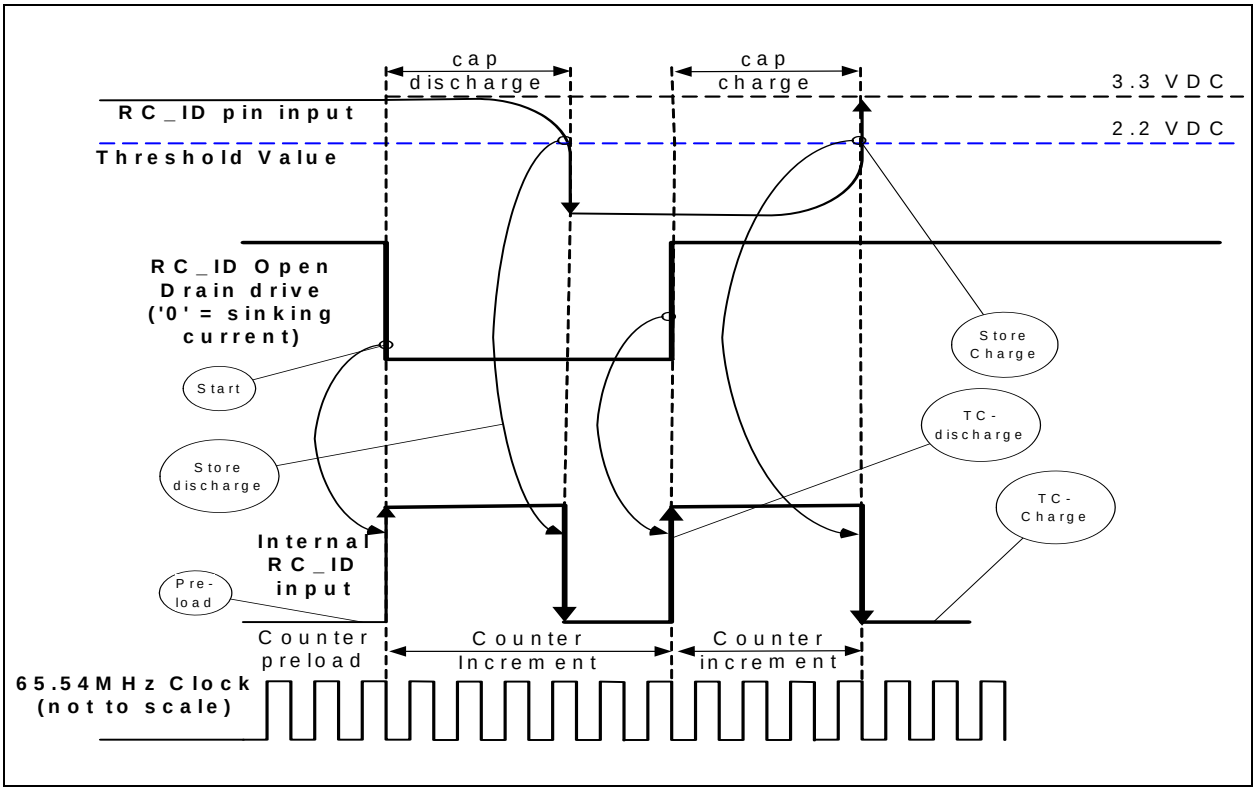
The RC_ID block initiates the discharging followed by the charging of the external RC circuit (see Figure 30-1. At the same time, the RCID input goes through a threshold detector set at 68% of 3.3 VDC. The TD_OUT gates the 16 bit Counter with a 125 ηs Resolution per bit with a 8.0 MHz clock or 62.5 ηs with a 16.0 MHz clock. The input has an input glitch rejection filter. Any change in input less than 210 ηs is ignored.

APPLICATION NOTE: After completion of a measurement cycle with the DONE bit set to '1' in the RCID_CTL Register, the programmer must place the RC_ID back into the reset state to before starting a new measurement.

TABLE 30-4: RC_ID MEASUREMENT STATES

State	Description
Reset	The RCID_CTL Register ENABLE bit is cleared to '0' by nSYS_RST or a write. The pin OD output driver is tristated, the RCID blocked is placed in low power mode. The DONE, CY_ER, TC bits in the RCID_CTL Register are autonomously cleared to '0'. **Note:** Clearing the ENABLE bit with a value other than 00h to the RCID_CTL Register is not defined and may create unpredictable results.
Enabled	Setting the ENABLE bit to '1' in the RCID_CTL Register places the RC_ID interface active high power state.
Start	The Start state is initiated by a write to the RCID_CTL Register setting the START bit to '1'. The counter is initiated to the preload value of 0000h and starts incrementing. The pin OD driver begins to sink current and external capacitor starts discharging. The DONE, CY_ER, TC bits in the RCID_CTL Register are autonomously cleared to '0'.
Detect Discharge	The pin voltage decays as the external capacitor discharges and the counter continues to increment until the terminal count is reached. The pin voltage is monitored to detect the discharge voltage reaches below the threshold voltage before the counter reaches terminal count.
TC-Discharged	The incrementing counter reaches the terminal count value of FFFFh. The TC bit in the RCID_CTL Register is autonomously set to '1'. If the pin voltage fails to discharge below the threshold voltage before the counter reaches terminal count during Detect Discharge, then CY_ER bit in the RCID_CTL Register is autonomously set to '1'; otherwise, then CY_ER bit remains clear. The pin OD output driver is tristated and the counter starts incrementing from 0000h.
Detect Charge	The pin voltage rises to the threshold voltage, counter stops counting, and the present value of the counter is stored in the RC_ID Data Register. The DONE bit in the RCID_CTL Register is autonomously set to '1'.
TC-Charged	The incrementing counter reaches the terminal count value of FFFFh and the pin voltage is below the threshold value. The CY_ER bit and DONE bits in the RCID_CTL Register are autonomously set to '1'. **Note:** The Detect Charge and the TC-Discharged STATES are mutually exclusive.

FIGURE 30-3: RC_ID TIMING DIAGRAMRC_ID OPERATION AND TIMINGS



All registers are VTR powered and are placed in reset when nSYS_RST (internal signal) is ‘0’.

30.7 Registers

Each instance of the RC Identification Detection (RC_ID) has its own Logical Device Number, and Base Address as indicated in Table 30-5.

TABLE 30-5: RC Identification Detection (RC_ID) BASE ADDRESS TABLE

RC ID Instance	LDN from (Table 3-2 on page 48)	AHB Base Address
RC ID	4h	F0_1000h

Table 30-6 is a register summary for this instance of the RC Identification Detection (RC_ID).

TABLE 30-6: RC Identification Detection (RC_ID) REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
RCID_CTL Register	00h	0	R/W	
RC_ID Data Register	04h	0	R	
		1		

30.8 Runtime Registers

30.8.1 RC_ID CONTROL REGISTER

TABLE 30-7: RCID_CTL REGISTER

HOST OFFSET	N/A						**HOST SIZE**	
EC OFFSET	00h					16-Bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						Clock_Sel	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R	R	R	R	R	R
BIT NAME	ENABLE	START	Reserved			CY_ER	TC	DONE

DONE

This read only status is set when the RCID completes a measurement and enters the “Detect Charge” or “TC-Charged” Measurement States described in Table 30-4, “RC_ID Measurement States,” on page 408.

This bit is cleared when RCID enters the “Reset” or “Start” Measurement States described in Table 30-4, “RC_ID Measurement States,” on page 408.

TC

This read only status bit is set when the RCID enters the “TC-Discharged” or “TC-Charged” Measurement State described in Table 30-4, “RC_ID Measurement States,” on page 408.

This bit is cleared when RCID enters the “Reset” or “Start” Measurement States described in Table 30-4, “RC_ID Measurement States,” on page 408.

CY_ER

This bit is a read only status bit and indicates when set to ‘1’ that the counter reached terminal count during the Capacitive Discharge or Charge phases without crossing the voltage threshold. This is an error condition.

START

Setting this bit to ‘1’ causes the RCID to enter the “Start” RCID Measurement State described in Table 30-4, “RC_ID Measurement States,” on page 408.

All writes to this register during other RC_ID states should clear this bit to ‘0’. See Note:.

ENABLE

Clearing this bit to ‘0’ causes the RCID to enter the “Reset” RCID Measurement State described in Table 30-4, “RC_ID Measurement States,” on page 408.

Setting this bit to ‘1’ starts the 12.904MHz clock input to the RCID and arms the counter.

APPLICATION NOTE: The ENABLE bit should remain set during the entire measurement; therefore all writes to the RCID_CTL Register during a measurement should set this bit.

Note:	When writing to the RCID_CTL Register to clear the ENABLE bit, the START bit should be cleared to '0'. Clearing the ENABLE bit with a value other than '0' is not defined and may create unpredictable results.

CLOCK_SEL

This field selects the frequency of the Counter circuit clock. Table 30-8, "Clock Select Field" shows the clock frequencies that can be selected:

TABLE 30-8: CLOCK SELECT FIELD

Clock_Sel	Counter Clock		Full Count Duration
0	DIVIDE BY 1	64.512 MHz	1.02 µsec
1	DIVIDE BY 2	32.256 MHz	2.03 µsec
2	DIVIDE BY 4	16.128 MHz	4.06 µsec
3	DIVIDE BY 8	8.064 MHz	8.13 µsec

The values in the Clock Select Field should only be changed when the ENABLE bit in the RCID_CTL Register is cleared to '0'.

30.8.2 RC_ID DATA REGISTER

The RC_ID Data Register provides a 16 bit counter value with a a 77.5 ns Resolution per bit.

Reads from this register in the Detect Charge Measurement States described in Table 30-4, "RC_ID Measurement States," on page 408 provides the bytes of the measured result for the Charge time.

TABLE 30-9: RC_ID DATA REGISTER

HOST ADDRESS							**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VTR			0000h			EC SPB **DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Data[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Data[7:0]							

30.9 Low Power Mode

The RC Identification Detection (RC_ID) interface is designed to conserve power when sleeping or disabled. Table 30-10 summarizes the RC Identification Detection (RC_ID) interface Low Power Mode behavior.

TABLE 30-10: BLOCK CLOCK GATING IN LOW POWER MODES

ENABLE **Bit**	DONE **Bit**	RCID_SLEEP_EN	**Block Idle Status (Note 30-1)**	RCID_CLOCK_REQ	**State**	**Description**
0	X	X	X	0	SLEEPING	The block is disabled and the clock can be stopped.
1	0	0	NOT IDLE	1	NORMAL OPERATION	The block is not idle and neither disabled by firmware nor commanded to sleep.
1	0	1	NOT IDLE	1	PREPARING TO SLEEP	The block is commanded to sleep, but the clock is required until the Block is idle.
1	1	1	IDLE	0	SLEEPING	The block is commanded to sleep and idle. The clock can be stopped.

Note 30-1 The DONE bit indicates the RC Identification Detection (RC_ID) interface 'idle' state.

31.0 GENERAL PURPOSE SERIAL PERIPHERAL INTERFACE (GP-SPI)

31.1 General Description

The SPI interfaces may be used to communicate with various peripheral devices, e.g., EEPROMS, DACs, ADCs, that use a standard Serial Peripheral Interface. There are two instances of GP-SPI controller, one located on the EC SPB bus and the other on the LPC SPB bus. The latter is intended for flash access by the host and EC; it can optionally work in conjunction with the DMA Controller to move data to and from the closely coupled SRAM with minimal software overhead.

Characteristics of the GP-SPI Controller include:

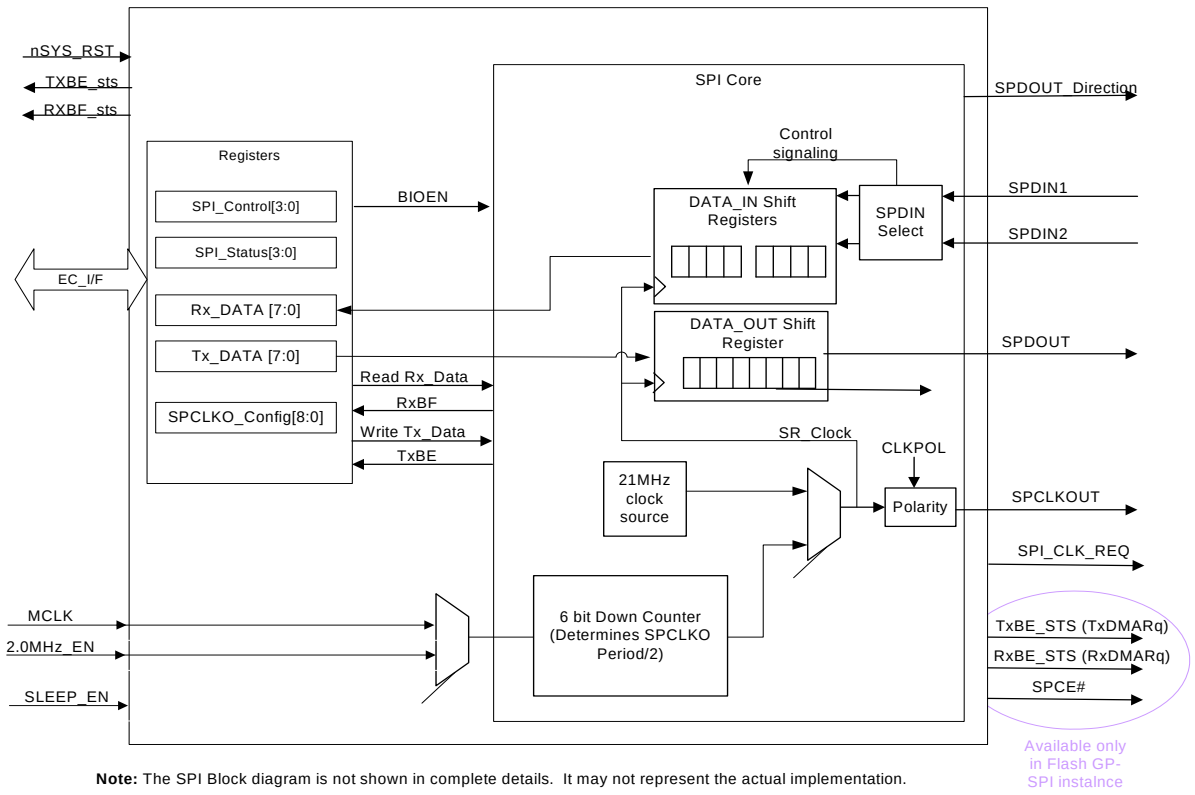
- 8-bit serial data transmitted and received simultaneously over two data pins in Full Duplex mode with options to transmit and receive data serially on one data pin in Half Duplex (Bidirectional) mode.
- An internal programmable clock generator and clock polarity and phase controls allowing communication with various SPI peripherals with specific clocking requirements.
- SPI cycle completion that can be determined by status polling or interrupts.
- The ability to read data in on both SPDIN and SPDOUT in parallel. This allows this SPI Interface to support dual data rate read accesses for emerging double rate SPI flashes
- Support of back-to-back reads and writes without clock stretching, provided the host can read and write the data registers within one byte transaction time.
- Hardware hooks to DMA Engine (available only in the Flash GP-SPI on the LPC SPB bus).

The MEC1609/MEC1609i SPI is a master only device and does not support multiple-master SPI configurations.

The GP-SPI controller on LPC SPB bus has its IO signals (pins) multiplexed with those from of the EC AHB SPI Flash Read Controller whose Master Bridge Enable register bit controls the multiplexer. See Section 2.4.19, "SPI Controllers Interface," on page 21 for pins listing.

31.2 SPI Block Diagram

FIGURE 31-1: SPI BLOCK DIAGRAM



Note: The SPI Block diagram is not shown in complete details. It may not represent the actual implementation.

31.3 SPI Block Diagram Signal Description

TABLE 31-1: SPI BLOCK SIGNALS

Signal	Direction	Description
nSYS_RST	INPUT	VTR Power on Reset.
TXBE_STS	OUTPUT	Interrupt output to EC driven by TXBE status bit
RXBF_STS	OUTPUT	Interrupt output to EC driven by RXBF status bit
64.52MHz	INPUT	Clock input to SPI Interface logic
2.0MHz_EN	INPUT	Clock Enable input to SPI Interface logic
SPDOUT	OUTPUT	Serial Data Out to the SPDOUT pin
SPDIN1	INPUT	Serial Data In 1 from SPPDIN pin. Input in full-duplex mode and Dual Read mode
SPDIN2	INPUT	Serial Data In 2 from SPDOUT pin. Input in bi-directional mode and Dual Read mode.
SPI_CLK	OUTPUT	SPI Clock output used to drive the SPCLK pin.

TABLE 31-1: SPI BLOCK SIGNALS (CONTINUED)

Signal	Direction	Description
SPDOUT_Direction	OUTPUT	The SPDOUT pin may be used as an output or an input. This signal is used to determine the direction of the SPDOUT buffer. 0=output (SPDOUT pin is controlled by the SPDOUT signal) 1=input (SPDOUT pin is an input driving the SPDIN2 signal) The SPDOUT pin has I/O capability. The I/O capability is implemented to support the Half-Duplex mode of operation (also referred to as bi-directional mode) and the Dual Read mode.
SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power Mode on page 418.
SPI_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= 64MHz can be turned 'off' when appropriate 1= 64MHz is required to be 'on.'
LPC/EC SPB Bus IF	I/O Bus	Bus used by microprocessor to access the registers in this block.
SPI_CS#	OUTPUT	SPI chip select (only for the Flash GP-SPI on LPC SPB bus)
TxBE_STS	OUTPUT	Tx DMA request (only for the Flash GP-SPI on LPC SPB bus)
RxBE_STS	OUTPUT	Rx DMA request (only for the Flash GP-SPI on LPC SPB bus)

31.4 SPI Interface Signals

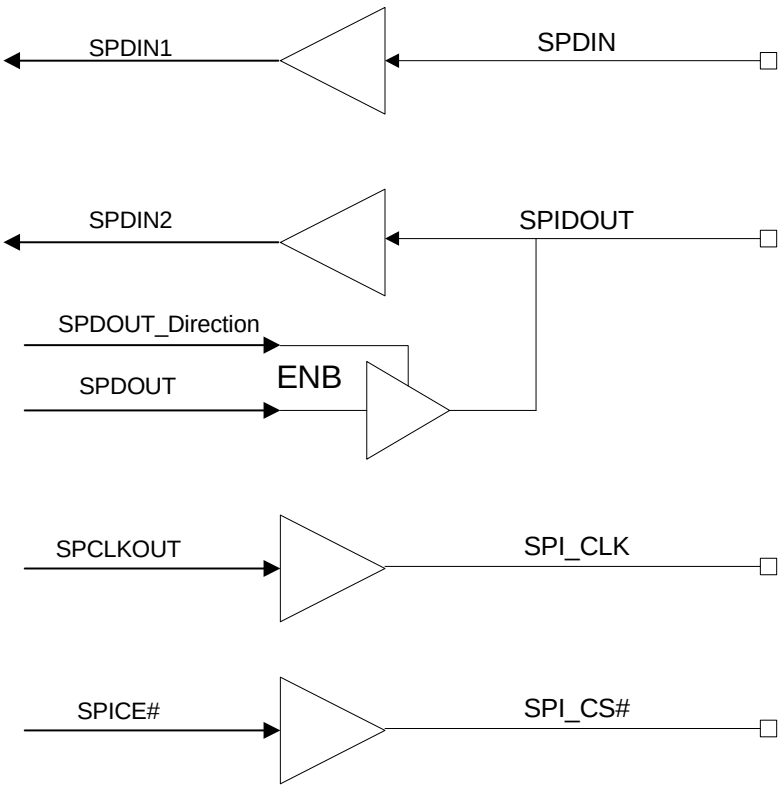
The following subsections describe the SPI Block Signals that are routed to the SPI pins. This chapter utilizes generic signal nomenclature for the Pin Signal Functions. Table 31-2 is a specific lookup table for pin signal function names used in this chapter and used elsewhere. Figure 31-2 show typical routing of the block interface (illustrated in Figure 31-1 to the pins. Table 2-15, “Miscellaneous Functions,” on page 17 in Pin Configuration includes the pin description for the SPI interface.

Since there is only one instance of SPI port in the MEC1609/MEC1609i, there is a one to one correspondence.

TABLE 31-2: PIN SIGNAL FUNCTION NOMENCLATURE LOOKUP TABLE

MEC1609 Pin Ref. Number	Routing Figure Generic Pin Signal Name	Pin Signal Function Name Used in other Chapter	Pin Function Signal Description
15	SPCLK	ECGP_SCLK	General Purpose SPI Clock
16	SPDOUT	ECGP_SOUT	General Purpose SPI Output
17	SPDIN	ECGP_SIN	General Purpose SPI Input

FIGURE 31-2: TYPICAL BLOCK/PIN INTERFACE



Note: This figure is for illustration purposes only, and not meant to imply specific implementation details

31.4.1 SPDOUT PIN - SERIAL PERIPHERAL DATA OUT

In Full Duplex Mode, this is the serial data output to the SPI interface. In half-duplex mode (also referred to as bi-directional mode) this is the serial data I/O port for the SPI interface.

For special SPI Flash devices that support Dual Read Modes the SPDOUT operates as in input in parallel with the SPDIN during the data portion of the Fast Dual Read command.

Note:	In the Bi-directional mode, some slave devices may tristate the last few bits to signal a turn-around; therefore, an external weak pull-up may be required on the pin.

31.4.2 SPDIN PIN - SERIAL PERIPHERAL DATA IN

In Full Duplex Mode, this is the serial data input from the SPI interface. In half-duplex mode (also referred to as bi-directional mode) this pin is unused.

Note:	Some slave device may tristate the SPDIN pin during command phase; therefore, an external weak pull-up or pull-down may be required on the pin.

31.4.3 SPCLK PIN - SERIAL PERIPHERAL CLOCK

This is the serial clock driven by the MEC1609/MEC1609i SPI (master) and connected to all SPI slaves. All data (input and output) is sampled/shifted on SPCLK according to the clock controls CLKPH and CLKPOL. (See TCLKPH and CLKPOL in Section 31.11.6, "SPICC - SPI Clock Control Register," on page 429.)

Note 31-1 In the MEC1609/MEC1609i, the General Purpose Serial Peripheral Interface (GP-SPI) pins are 8 mA buffers. The maximum SPCLK pin clock frequency is 16.128MHz for all modes. Limited functionality is available at 32.26 MHz and although the block can be programmed for higher frequencies performance may not be maintained. See TABLE 31-14: on page 430 and Section 31.9.5.5, "Limits of SPI configurations," on page 424.

31.5 Power, Clocks and Reset

31.5.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

31.5.2 CLOCKS

This block uses the EC Bus Clock / LPC Bus Clock, the 64.52MHz (MCLK) and the 2MHz (MCLK_DIV32_EN). EC Bus Clock / LPC Bus Clock is used when reading and writing the General Purpose Serial Peripheral Interface (GP-SPI) registers. The 6-bit down counter may use either MCLK or MCLK_DIV32_EN to directly decrement the counter, which is the SPI_CLK source.

See Section 5.4, "Clock Generator," on page 76 for definition and details on clocks of: MCLK on page 91, MCLK_DIV32_EN on page 91, and EC Bus Clock on page 92.

31.6 Reset

This block is reset on a nSYS_RST. On reset, the General Purpose SPI interface defaults to disabled, The block can also be reset by software, by setting the Soft Reset bit located in the SPICR - SPI Control Register. Setting this bit reinitializes the SPI Control block back to its nSYS_RST state.

See Section 5.6, "Reset Interface," on page 95 for definition and details on reset: nSYS_RST on page 98.

31.7 SPI Interrupts / DMA Requests

The General Purpose Serial Peripheral Interface (GP-SPI) can generate an interrupt events to the Embedded Controller (EC) to indicate that the block requires servicing. The SPI TXBE status and RXBF status bits in the SPISR - SPI Status Register on page 427 are routed onto the SPI_TXBE_GP & SPI_RXBF_GP bits of the GIRQ14 Source Register on page 274. In the Flash GP-SPI instance, these status bits are also connected respectively to the DMA Controller's SPI Flash Write and Read requests signals.

31.8 Low Power Mode

This block is designed to conserve power when it is either sleeping or disabled. There are two ways to put the SPI interface into a low power mode: Disabled the SPI Interface via the Enable Bit or Assert the SLEEP_EN signal to the SPI Interface. The following table summarizes the SPI behavior for each of these low power modes.

TABLE 31-3: BLOCK CLOCK GATING IN LOW POWER MODES

Enable Bit	SLEEP_EN	Block Idle Status	SPI_CLK_REQ	State	Description
0	X	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.
1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

31.8.1 DISABLING THE SPI INTERFACE BLOCK VIA THE ENABLE BIT

The enable bit is located in Section 31.11.1, "SPIAR - SPI Enable Register," on page 426. When this bit is cleared the SPI interface is in its lowest power state. The 64.52MHz clock input is gated and the SPDOUT and SPI_CLK pins are set to their inactive state as determined by the configuration bits.

Note: The SPI Interface is required to finish the current transaction and enter the Idle state before deasserting the SPI_CLK_REQ signal and gating its internal clock source.

31.8.2 ASSERTING THE SLEEP_EN SIGNAL TO THE SPI INTERFACE BLOCK

When the SLEEP_EN signal is asserted the SPI interface completes the current transaction and then enters the low power state. In the low power state the 64.52MHz clock input is gated, the SPDOUT and SPI_CLK pins are set to their inactive state as determined by the configuration bits, and the SPI_CLK_REQ signal is de-asserted.

31.9 Operation

The Serial Peripheral Interface (SPI) block is a master SPI block used to communicate with external SPI devices. The SPI master is responsible for generating the SPI clock and is designed to operate in Full Duplex, Half Duplex, and Dual modes of operation. The clock source may be programmed to operated at various clock speeds. The data is transmitted serially via 8-bit transmit and receive shift registers. Communication with SPI peripherals that require transactions of varying lengths can be achieved with multiple 8-bit cycles.

This block has many configuration options: The data may be transmitted and received either MSbit or LSbit first; The SPI Clock Polarity may be either active high or active low; Data may be sampled or presented on either the rising of falling edge of the clock (referred to as the transmit clock phase); and the SPI_CLK SPDOUT frequency may be programmed to select values from 9.52kHz to 64.52MHz. In addition to these many programmable options, this feature has several status bits that may be enabled to notify the host that data is being transmitted or received.

31.9.1 INITIATING AN SPI TRANSACTION

All SPI transactions are initiated by a write to the TX_DATA register. No read or write operations can be initiated until the Transmit Buffer is Empty, which is indicated by a one in the TXBE status bit.

If the transaction is a write operation, the host writes the TX_DATA register with the value to be transmitted. Writing the TX_DATA register causes the TXBE status bit to be cleared, indicating that the value has been registered. If empty, the SPI Core loads this TX_DATA value into an 8-bit transmit shift register and begins shifting the data out. Loading the value into the shift register causes the TXBE status bit to be asserted, indicating to software that the next byte can be written to the TX_DATA register.

If the transaction is a read operation, the host initiates a write to the TX_DATA register in the same manner as the write operation. Unlike the transmit command, the host must clear the RXBF status bit by reading the RX_DATA register before writing the TX_DATA register. This time, the host will be required to poll the RXBF status bit to determine when the value in the RX_DATA register is valid.

Note 1: If the SPI interface is configured for Half Duplex mode, the host must still write a dummy byte to receive data.

2: Since RX and TX transactions are executed by the same sequence of transactions, data is always shifted into the RX_DATA register. Therefore, every write operation causes data to be latched into the RX_DATA register and the RXBF bit is set. This status bit should be cleared before initiating subsequent transactions. The host utilizing this SPI core to transmit SPI Data must discard the unwanted receive bytes.

3: The length and order of data sent to and received from a SPI peripheral varies between peripheral devices. The SPI must be properly configured and software-controlled to communicate with each device and determine whether SPIRD data is valid slave data.

The following diagrams show sample single byte and multi-byte SPI Transactions.

FIGURE 31-3: SINGLE BYTE SPI TX/RX TRANSACTIONS (FULL DUPLEX MODE)

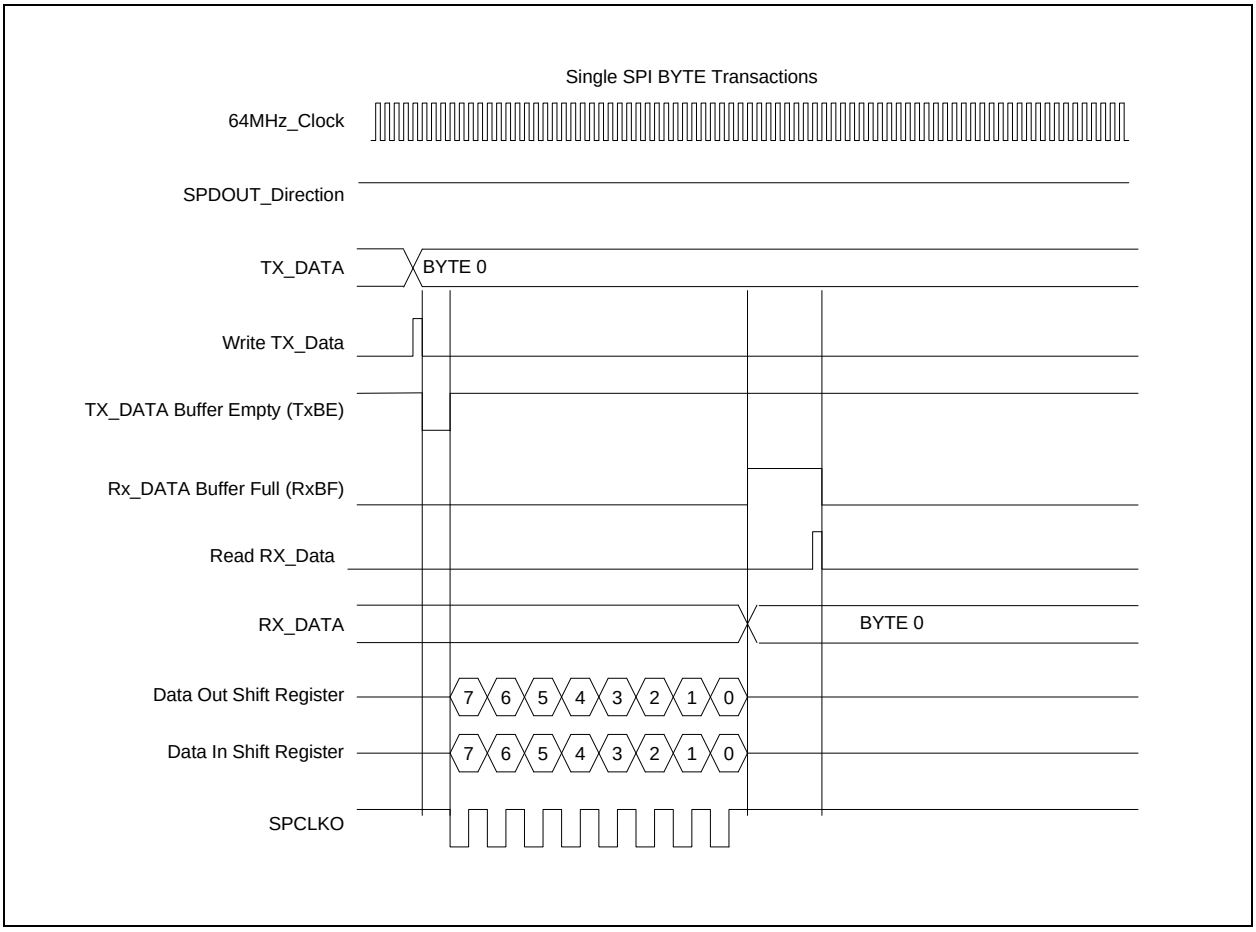
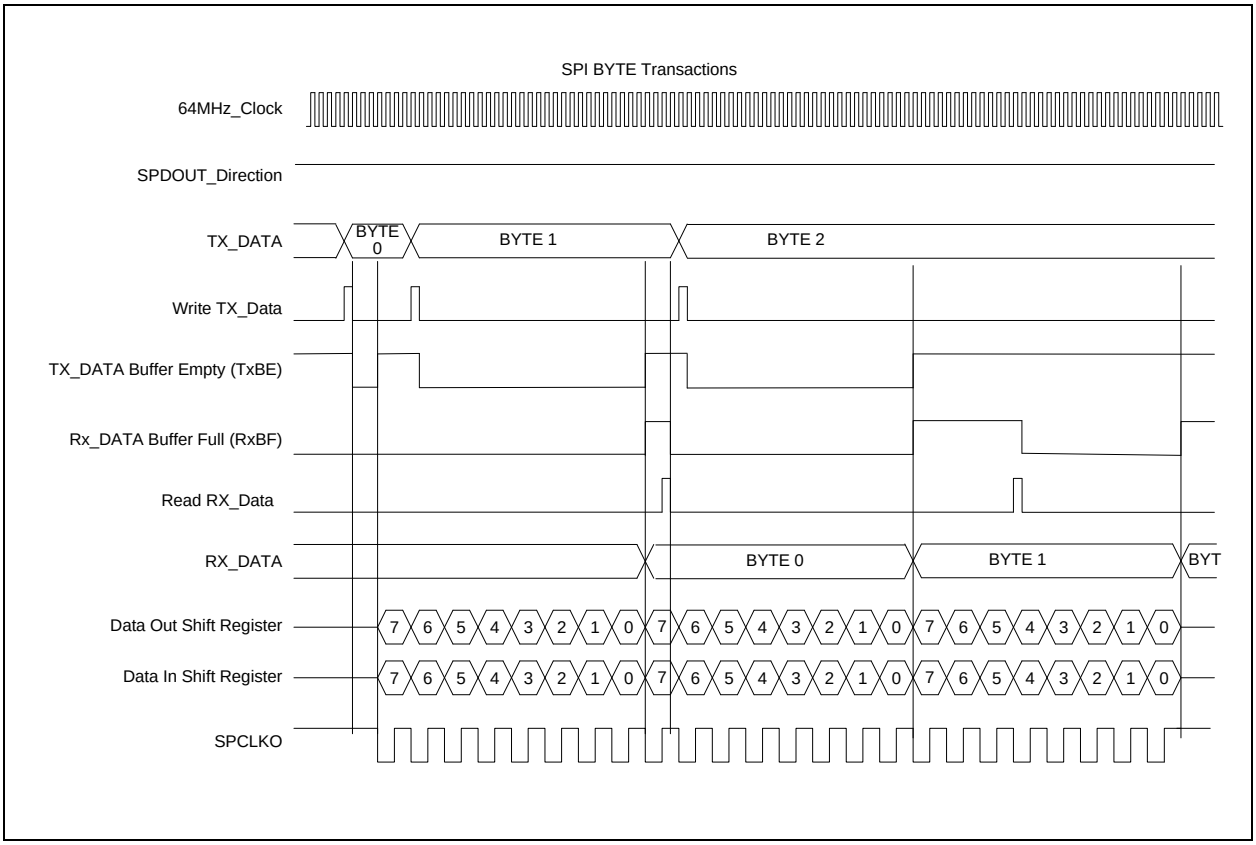


FIGURE 31-4: MULTI-BYTE SPI TX/RX TRANSACTIONS (FULL DUPLEX MODE)



The data may be configured to be transmitted MSB or LSB first. This is configured by the LSBF bit in the Section 31.11.2, "SPICR - SPI Control Register," on page 426. The transmit data is shifted out on the edge as selected by the TCLKPH bit in the SPICC register. See Section 31.11.6, "SPICC - SPI Clock Control Register," on page 429. All received data can be sampled on a rising or falling SPI_CLK edge using RCLKPH (see RCLKPH in Section 31.11.6, "SPICC - SPI Clock Control Register," on page 429 for clock controls). This clock setting must be identical to the clocking requirements of the current SPI slave.

Note: Common peripheral devices require a chip select signal to be asserted during a transaction. Chip selects for SPI devices may be controlled by MEC1609/MEC1609i GPIO pins.

There are three types of transactions that can be implemented for transmitting and receiving the SPI data. They are Full Duplex, Half Duplex, and Dual Mode. These modes are define in Section 31.9.3, "Types of SPI Transactions," on page 421.

31.9.2 DMA MODE (FLASH GP-SPI ONLY)

Transmit and receive operations can use a DMA channel. Note that only one DMA channel may be enabled at a time. Setting up the DMA Controller involves specifying the device (Flash GP-SPI), direction (transmit/receive), and the start and end addresses of the DMA buffers in the closely couple memory. Please refer to the DMA Controller chapter for register programming information.

SPI transmit / DMA write: the GP-SPI block's transmit empty (TxBE) status signal is used as a write request to the DMA controller, which then fetches a byte from the DMA transmit buffer and writes it to the GP-SPI's SPI TX Data Register (SPITD). As content of the latter is transferred to the internal Tx shift register from which data is shifted out onto the SPI bus bit by bit, the Tx Empty signal is again asserted, triggering the DMA fetch-and-write cycle. The 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.

SPI receive / DMA read: the AUTO_READ bit in the SPI Control Register must be set. The driver first writes (dummy data) to the SPI TX Data Register (SPITD) to initiate the toggling of the SPI clock, enabling data to be shifted in. After one byte is received, the Rx Full (RxBF) status signal, used as a read request to the DMA controller, is asserted. The 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 afore-mentioned 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.

31.9.3 TYPES OF SPI TRANSACTIONS

The MEC1609/MEC1609i SPI can be configured to operate in three modes: Full Duplex, Half Duplex, and Dual Mode.

31.9.3.1 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.

31.9.3.2 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 (See Section , "BIOEN," on page 426).

- 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.

31.9.3.3 Dual Mode of Operation

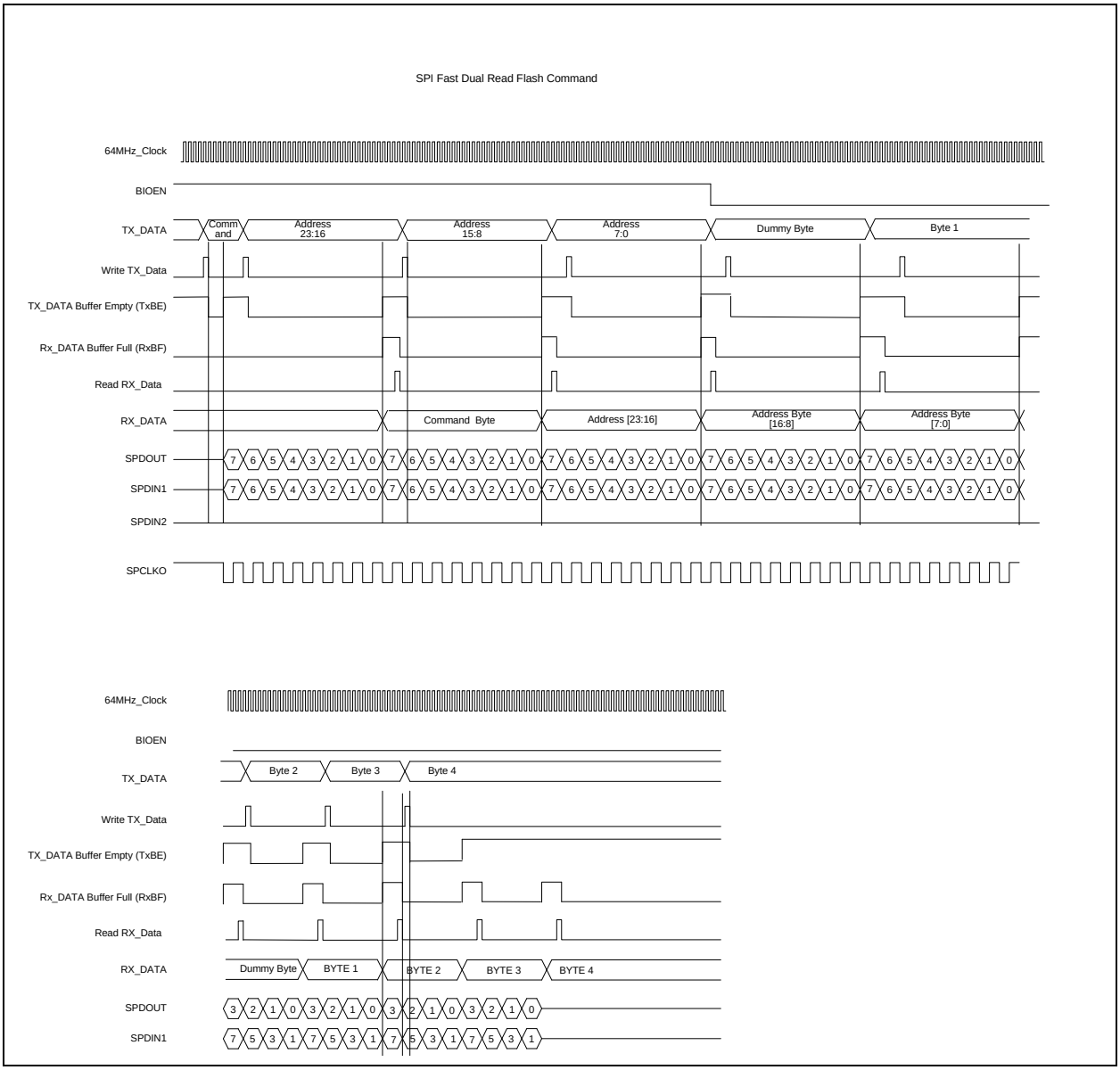
Note:	The Dual Mode of Operation has been implemented to support selected SPI Flash devices that support the Fast Dual Mode command.

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. 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. For a description of the BIOEN bit see BIOEN on page 426.

- 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.

FIGURE 31-5: DUAL FAST READ FLASH COMMAND



Note: When the SPI core is used for flash commands, like the Dual Read command, the host discards the bytes received during the command, address, and dummy byte portions of the transaction.

31.9.4 HOW BIOEN BIT CONTROLS DIRECTION OF SPDOUT BUFFER

When the SPI is configured for Half Duplex mode or Dual Mode the SPDOUT pin operates as a bi-directional signal. The BIOEN bit is used to determine the direction of the SPDOUT buffer when a byte is transmitted. Internally, the BIOEN bit is sampled to control the direction of the SPDOUT buffer when the TX_DATA value is loaded into the transmit shift register. The direction of the buffer is never changed while a byte is being transmitted.

Since the TX_DATA register may be written while a byte is being shifted out on the SPDOUT pin, the BIOEN bit does not directly control the direction of the SPDOUT buffer. An internal DIRECTION bit, which is a latched version of the BIOEN bit determines the direction of the SPDOUT buffer. The following list summarizes when the BIOEN bit is sampled.

- The DIRECTION bit is equal to the BIOEN bit when data is not being shifted out (i.e., SPI interface is idle).
- The hardware samples the BIOEN bit when it is shifting out the last bit of a byte to determine if the buffer needs to be turned around for the next byte.
- The BIOEN bit is also sampled any time the value in the TX_DATA register is loaded into the shift register to be transmitted.

APPLICATION NOTE: If a TAR (Turn-around time) is required between transmitting and receiving bytes on the SPDOUT signal, software should allow all the bytes to be transmitted before changing the buffer to an input and then load the TX_DATA register to begin receiving bytes. This allows the SPI block to operate the same as legacy SPI devices.

31.9.5 CONFIGURING THE CLOCK GENERATOR FOR AN SPI TRANSACTION

The SPI Core generates the SPI_CLK signal to the external SPI device. This clock may be configured for frequencies from 9.52kHz to 64.52MHz. The clock phase and polarity are configurable as well. The following sections define how to program these features.

USER'S NOTE: The clock source configuration should not be changed during an SPI transaction.

31.9.5.1 Configuring the Frequency of the SPI Clock

The frequency of the SPI_CLK signal is determined by the clock source enabled to the clock generator and the preload value of the clock generator down counter. The clock generator toggles the SPI_CLK output every time the counter underflows, while data is being transmitted. If the preload value is set to 0 the 64.52 MHz clock source bypasses the down counter to directly create the clock generator output.

Note: When the SPI interface is in the idle state and data is not being transmitted, the SPI_CLK signal stops in the inactive state as determined by the configuration bits.

The clock source to the down counter is determined by Bit CLKSRC. Either the 64.52MHz clock or the 2.0MHz clock enable can be used to decrement the down counter in the clock generator logic.

31.9.5.2 Configuring the SPI Mode

In practice, there are four modes of operation that define when data should be latched. These four modes are the combinations of the SPI_CLK polarity (CLKPOL) and phase (RCLKPH and TCLKPH). Phase is programmable independently for the receive and transmit phases. CLKPOL, RCLKPH and TCLKPH bits are programmable as defined in Section 31.9.5.3 and Section 31.9.5.4 below.

TABLE 31-4: SPI MODES

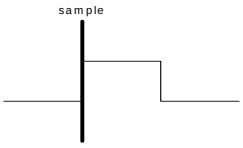
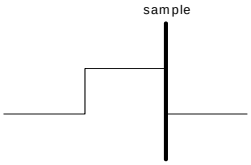
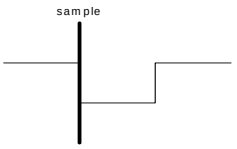
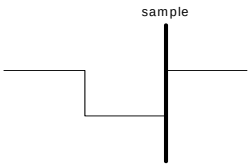
SPI Mode	CLKPOL	CLKPH	Definition	Diagram
0	0	0	data sampled on rising edge of clock	
1	0	1	data sampled on falling edge of clock	

TABLE 31-4: SPI MODES (CONTINUED)

SPI Mode	CLKPOL	CLKPH	Definition	Diagram
2	1	0	data sampled on falling edge of clock	
3	1	1	data sampled on rising edge of clock	

31.9.5.3 Configuring the Polarity of the SPI Clock

The output of the clock generator may be inverted to create an active high or active low clock pulse. This is used to determine the inactive state of the SPI_CLK signal and is used for determining the first edge for shifting the data. The polarity is selected by Bit CLKPOL in the SPI Clock Control Register (SPICC).

31.9.5.4 Configuring the Phase of the SPI Clock

The SPI devices need to know when to sample the data, which may be either on the first edge of the clock or on the second edge of the clock. The phase of the clock is selected independently for receiving data and transmitting data. The receive phase is determine by Bit RCLKPH and the transmit phase is determine by TCLKPH in the SPI Clock Control Register (SPICC).

31.9.5.5 Limits of SPI configurations

The following limits Modes, clock frequency, & board layout.

Note 31-2 In the MEC1609/MEC1609i, the General Purpose Serial Peripheral Interface (GP-SPI) pins are 8 mA buffers. The maximum SPCLK pin clock frequency is 16.128MHz for all modes. Limited functionality is available at 32.26 MHz and although the block can be programmed for higher frequencies performance may not be maintained. See TABLE 31-14: on page 430 and Section 31.9.5.5, "Limits of SPI configurations," on page 424.

Master (MEC1609/MEC1609i)				Slave (Not MEC1609/MEC1609i)	
Max SPICLK Frequency	Dual Mode of Operation	**Output Data Transitions**	**Input Data Sampled**	**Output Data Transitions**	**Input Data Sampled**
64.52MHz	Not supported				
32.26MHz	Not supported	Pos edge of clk	Pos edge of clk	Pos edge of clk	Either edge of clk
16.13MHz or less	Supported	All combinations are valid			

31.10 Instance Description

There are two instances of General Purpose Serial Peripheral Interface (GP-SPI) block implemented in the MEC1609/MEC1609i.

Each instance of the General Purpose Serial Peripheral Interface (GP-SPI) has its own Logical Device Number, and Base Address as indicated in Table 31-5.

TABLE 31-5: General Purpose Serial Peripheral Interface (GP-SPI) BASE ADDRESS TABLE

General Purpose Serial Peripheral Interface (GP-SPI) Instance	LDN from (Table 3-2 on page 48)	AHB Base Address
EC GP-SPI	7h	F0_1C00h
Flash SPI	Fh	FF_3C00h (Note 31-3)

The Table 31-6 is a register summary for one instance of the General Purpose Serial Peripheral Interface (GP-SPI). Each EC address is indicated as an SPB Offset from its AHB base address.

Note 31-3 All of the addresses in the Detailed Register Descriptions below refer to the EC GP-SPI instance register addresses which are 32-bit aligned and must be divided by four to correctly represent the addressing for the Flash SPI instance which is accessible to the LPC host.

TABLE 31-6: General Purpose Serial Peripheral Interface (GP-SPI) REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
SPIAR - SPI Enable Register	00h	3-0	R/W	
SPICR - SPI Control Register	04h	3-0	R/W	
SPISR - SPI Status Register	08h	3-0	R	
SPITD - SPI TX_Data Register	0Ch	3-0	R/W	
SPIRD - SPI RX_Data Register	10h	3-0	R	
SPICC - SPI Clock Control Register	14h	3-0	R/W	
SPICG - SPI Clock Generator Register	18h	3-0	R/W	
SPIAR - SPI Enable Register	00h	3-0	R/W	

31.11 Detailed Register Descriptions

APPLICATION NOTE: In the SPI registers some configuration bits are assumed to be static, while others may be updated dynamically by software. The BIOEN and ENABLE bits are considered dynamic bits that can be modified by software at anytime, regardless if a transaction is active or not. These values are latched in hardware, so as to not affect the current operation being performed. All other bits are considered static and cannot be changed by Software while an SPI Transaction is in process.

31.11.1 SPIAR - SPI ENABLE REGISTER

TABLE 31-7: SPI ENABLE REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	00h				8-bit		**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Enable

ENABLE

0=Disabled. Clocks are gated to conserve power and the SPDOUT and SPI_CLK signals are set to their inactive state

1=Enabled. The device is fully operational.

31.11.2 SPICR - SPI CONTROL REGISTER

TABLE 31-8: SPI CONTROL REGISTER

HOST ADDRESS	n/a				8-bit		**HOST SIZE**	
EC OFFSET	04h				8-bit		**EC SIZE**	
POWER	VTR				02h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	CE	AUTO_READ	Soft Reset	SPDIN Select		BIOEN	LSBF

LSBF

Least Significant Bit First control.

0= The data is transferred in MSB-first order. (default)

1= The data is transferred in LSB-first order.

BIOEN

Bidirectional Output Enable control. When the SPI is configured for Half Duplex mode or Dual Mode the SPDOUT pin operates as a bi-directional signal. The BIOEN bit is used by the internal DIRECTION bit to control the direction of the SPDOUT buffers. The direction of the buffer is never changed while a byte is being transmitted.

0=The SPDOUT_Direction signal configures the SPDOUT signal as an input.

1=The SPDOUT_Direction signal configures the SPDOUT signal as an output. (default)

Note:	If the SPIMODE bit is configured for Full Duplex mode the BIOEN bit must be set to ‘1’ to configure the SPDOUT signal as an output.

APPLICATION NOTE: Although the design supports back-to-back transmissions even when the direction of the buffer is changed, it is the software's responsibility to avoid collisions on the SPDOUT signal. The designed has been implemented to support a 0 second (max) turn-around (TAR) time. If TAR greater than zero is required, the software must wait for the transmission in one direction to complete before writing the TX_DATA register to start sending/receiving in the opposite direction.

SPDIN SELECT

The SPDIN Select which SPI input signals are enabled when the BIOEN bit is configured as an input.

00= SPDIN1 only //Select this option for Full Duplex (default)

01=SPDIN2 only //Select this option for Half Duplex

1x=SPDIN1 and SPDIN2 //Select this option for Dual Mode

SOFT RESET

Soft Reset is a self-clearing bit. Writing zero to this bit has no effect. Writing a one to this bit resets the entire SPI Interface, including all counters and registers back to their initial state (i.e., the same as a nSYS_RST).

AUTO_READ

When this bit is 1, a read of the SPI RX_DATA Register will both clear the RXBF status bit, and in addition it will clear the TXBE status bit. Clearing the TXBE status bit will cause the contents of the TX_DATA register to be copied into the 8-bit transmit shift register, which then begins shifting data out. Because shifting data out is accompanied by clocking the SPI Clock, a read of RX_DATA will cause the next byte from the SPI device to be shifted in to RX_DATA. The contents of the TX_DATA register is not significant, since when reading from RX_DATA it is just used to generate the SPI clock. When this bit is 0 (default), a read of the SPI RX_DATA Register will clear the RXBF bit, but not the TXBE bit.

CE

1= SPCE# output signal is asserted, i.e., driven to logic '0'

0= SPCE# output signal is asserted, i.e., driven to logic '1'

31.11.3 SPISR - SPI STATUS REGISTER

TABLE 31-9: SPI STATUS REGISTER

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	08h					8-bit	**EC SIZE**	
POWER	VTR					01h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved					ACTIVE	RxBF	TxBE

TXBE

Transmit Data Buffer Empty status. This bit is a read-only bit used to indicate that the Tx_Data buffer is empty. Writing the Tx_DATA Buffer clears this bit. This signal may be used to generate an interrupt to the EC, if not masked.

RXBF

Receive Data Buffer Full status. This bit is a read-only bit used to indicate when the Rx_Data buffer is full. Reading the Rx_DATA Buffer clears this bit. This signal may be used to generate an interrupt to the EC, if not masked.

ACTIVE

The ACTIVE bit indicates that a transaction controlled by the General Purpose Serial Peripheral Interface (GP-SPI) is in progress. The ACTIVE bit is asserted ('1') when the GP-SPI controller transfers data from the SPITD - SPI TX_Data Register into the DATA_OUT Shift Register.

ACTIVE is de-asserted ('0') follow a system reset (nSYS_RST) or when data from the DATA_IN Shift Register is transferred to the SPIRD - SPI RX_Data Register. Note that data is only transferred from the DATA_IN Shift Register to the SPIRD - SPI RX_Data Register when RxBF is not asserted ('0'); i.e., the SPIRD - SPI RX_Data Register is not overwritten by incoming data when RxBF is '1.'

31.11.4 SPITD - SPI TX_DATA REGISTER

TABLE 31-10: SPI TX DATA REGISTER (SPITD)

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	0Ch					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	TX_Data[7:0]							

TX_DATA[7:0]

A write to this register with the TxBE bit asserted '1' initiates an SPI transaction. If the Transmit Shift Register is empty the byte written to this register will be loaded into the shift register and the TxBE flag will be asserted. This indicates that the next byte can be written into the TX_DATA register. This byte will remain in the TX_DATA register until the SPI core has finished shifting out the previous byte. Once the shift register is empty, the hardware will load the pending byte into the shift register and once again assert the TxBE bit.

Note:	The TX_DATA register must not be written when the TxBE bit is zero. Writing this register may overwrite the transmit data before it is loaded into the shift register.

Reading the TX_DATA register will return the last value written to this register.

31.11.5 SPIRD - SPI RX_DATA REGISTER

TABLE 31-11: SPI RX DATA REGISTER (SPIRD)

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	10h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	RX_Data[7:0]							

RX_DATA[7:0]

This register is used to read the value returned by the external SPI device. At the end of a byte transfer the RX_DATA register contains serial input data (valid or not) from the last transaction and the RXBF bit is set to one. This status bit indicates that the RX_DATA register has been loaded with a the serial input data. The RX_DATA register should not be read before the RXBF bit is set.

Note: The RX_DATA register must be read, clearing the RXBF status bit before writing the TX_DATA register. The data in the receive shift register is only loaded into the RX_DATA register when this bit is cleared. If a data byte is pending in the receive shift register the value will be loaded immediately into the RX_DATA register and the RXBF status flag will be asserted. Software should read the RX_DATA register twice before starting a new transaction to make sure the RX_DATA buffer and shift register are both empty.

31.11.6 SPICC - SPI CLOCK CONTROL REGISTER

TABLE 31-12: SPI CLOCK CONTROL REGISTER (SPICC)

HOST ADDRESS	n/a			8-bit			**HOST SIZE**	
EC OFFSET	14h			8-bit			**EC SIZE**	
POWER	VTR			02h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W	R	R/W	R/W	R/W
BIT NAME	Reserved			CLKSRC	Reserved	CLKPOL	RCLKPH	TCLKPH

APPLICATION NOTE: the default CLKPOL, RCLKPH and TCLKPH values are appropriate for transactions with typical MODE 0 SPI Flash devices.

TCLKPH

The TCLKPH bit determines the Transmit Clock Phase, the SPCLK edge on which the master will clock data out.

0= Valid data is clocked out on the SPDOUT signal prior to the first SPI_CLK edge. The slave device should sample this data on the first and following odd SPI_CLK edges (i.e., sample data on rising edge). (default)

1= Valid data is clocked out on the first SPI_CLK edge on SPDOUT signal. The slave device should sample this data on the second and following even SPI_CLK edges (i.e., sample data on falling edge).

Note: This functionality is independent of the polarity of SPCLK. See Section 41.10, "Serial Peripheral Interface (SPI) Timings," on page 517 for timing diagrams.

RCLKPH

The RCLKPH bit determines the Receive Clock Phase, the SPI_CLK edge on which the master will sample data.

0= Valid data is expected on the SPDIN signal on the first SPI_CLK edge. This data is sampled on the first and following odd SPI_CLK edges (i.e., sample data on rising edge).

1= Valid data on SPDIN signal is expected after the first SPI_CLK edge. This data is sampled on the second and following even SPI_CLK edges (i.e., sample data on falling edge). (default).

Note: This functionality is independent of the polarity of SPI_CLK. See Section 41.10, "Serial Peripheral Interface (SPI) Timings," on page 517 for timing diagrams.

CLKPOL

This bit controls the polarity of the SPI clock.

0= The SPI_CLK is low when the interface is idle and the first clock edge is a rising edge. (default)

1= The SPI_CLK signal is high when the interface is idle and the first clock edge is a falling edge

CLKSRC

This bit controls the clock source to SPI Clock Generator

0= The clock source to the SPI Clock Generator 6-bit down counter is 64.52MHz clock enable (default)

1= The clock source to the SPI Clock Generator 6-bit down counter is 2MHz clock enable

Note:	The CLKSRC bit should not be changed during a SPI transaction.

31.11.7 SPICG - SPI CLOCK GENERATOR REGISTER

TABLE 31-13: SPI CLOCK GENERATOR REGISTER (SPICG)

HOST ADDRESS	n/a				8-bit		**HOST SIZE**	
EC OFFSET	18h				8-bit		**EC SIZE**	
POWER	VTR				02h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		Preload[5:0]					

PRELOAD[5:0]

The SPI Clock Generator Preload value. The SPI_CLK signal is a clock output with a 50% duty cycle. The signal is generated from a 6-bit down counter that toggles the SPI_CLK pin every time it reaches zero and reloads the counter. This counter is decremented at the rate of the input clock enable, which is either the 64.52MHz clock enable or the 2MHz clock enable.

$$\text{SPCLKO\_FREQ}=\left(\left(\frac{1}{2}\times \text{CLOCK\_ENABLE\_FREQ}\right)/\text{PRELOAD}\right)$$

The PRELOAD field contains the preload value for the counter that determines the resulting frequency. The following table outlines the ranges of frequencies possible. Notice that a preload value of zero effectively bypasses the counter and maps the 64.52MHz clock onto the SPI_CLK signal.

TABLE 31-14: SPI_CLK FREQUENCIES

Clock Enable Frequency	Preload	SPI_CLK Frequency	Notes
Don't Care	0	64.52MHz	Max Frequency (Note 31-4)
64.52MHz	1	32.26MHz	
64.52MHz	2	16.13MHz	Default Frequency
64.52MHz	3	10.75MHz	
64.52MHz	63	0.51MHz	
2.016MHz	0	2.016MHz	
2.016MHz	1	1.008MHz	

TABLE 31-14: SPI_CLK FREQUENCIES (CONTINUED)

Clock Enable Frequency	Preload	SPI_CLK Frequency	Notes
2.016MHz	2	504kHz	
2.016MHz	3	336kHz	
2.016MHz	63	16kHz	Min Frequency

Note 31-4 When the Preload value is programmed to zero the 64.52MHz Clock Source is directly mapped to the SPI_CLK signal. Since the 64.52MHz signal is a 50% duty cycle it may be directly used as an SPI_CLK frequency.

Note 31-5 In the MEC1609/MEC1609i, the General Purpose Serial Peripheral Interface (GP-SPI) pins are 8 mA buffers. The maximum SPCLK pin clock frequency is 16.128MHz for all modes. Limited functionality is available at 32.26 MHz and although the block can be programmed for higher frequencies performance may not be maintained. See TABLE 31-14: on page 430 and Section 31.9.5.5, "Limits of SPI configurations," on page 424.

31.12 SPI Examples

31.12.1 FULL DUPLEX MODE TRANSFER EXAMPLES

31.12.1.1 Read Only

The slave device used in this example is a MAXIM MAX1080 10 bit, 8 channel ADC:

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is de-asserted '0' to enable the SPI interface in Full Duplex mode.
- The CLKPOL and TCLKPH bits are de-asserted '0', and RCLKPH is asserted '1' 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.
- Assert #CS using a GPIO pin.
- Write a valid command word (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 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.
- A dummy 8 bit data value (any value) 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 transmit bytes:
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit command data to the slave. This particular slave device drives '0' on the SPDIN pin to the master while it is accepting command data. This SPIRD data is ignored.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register 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.
- The final SPI cycle is initiated when another dummy 8 bit data value (any value) is written to the TX_DATA register. Note that this value may be another dummy value or it can be a new 8 bit command to be sent to the ADC. The new command will be transmitted while the final data from the last command is received simultaneously. This overlap allows ADC data to be read every 16 SPCLK cycles after the initial 24 clock cycle.The SPI master auto-

matically 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, the second SPI cycle is complete:
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is the first half of a valid 16 bit ADC value. SPIRD is read and stored.
  - Once the second SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- After 8 SPI_CLK pulses, the final SPI cycle is complete, TXBF is asserted '1', and the SPINT interrupt is asserted (if enabled). The data now contained in SPIRD - SPI RX_Data Register is the second half of a valid 16 bit ADC value. SPIRD is read and stored.
- If a command was overlapped with the received data in the final cycle, #CS should remain asserted and the SPI master will initiate another SPI cycle. If no new command was sent, #CS is released and the SPI is idle.

31.12.1.2 Read/Write

The slave device used in this example is a Fairchild NS25C640 FM25C640 64K Bit Serial EEPROM. The following sub-sections describe the read and write sequences.

Read

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is de-asserted '0' to enable the SPI interface in Full 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.
- Assert CS# low using a GPIO pin.
- Write a valid command word (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. 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.
- Next, EEPROM address A15-A8 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 transmit byte (Command Byte transmitted):
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit command data to the slave. This particular slave device tri-states the SPDIN pin to the master while it is accepting command data. This SPIRD data is ignored.

USER'S NOTE:	External pull-up or pull-down is required on the SPDIN pin if it is tri-stated by the slave device. Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A15-A8) 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. Note: The particular slave device ignores address A15-A13.

- 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, EEPROM address A7-A0 is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, 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 from the second transmit byte (Address Byte (MSB) transmitted):
  - EEPROM address A15-A8 has been transmitted to the slave completing the second SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). 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 second SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A7-A0) 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.
- Next, a dummy byte is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, 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 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. (See Section 31.7, "SPI Interrupts / DMA Requests"). 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 (dummy byte) 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 receive byte is required, the host would not write any more value to the TX_DATA register until this transaction completes. If more than one byte of data is to be received, another dummy byte would be written to the TX_DATA register (one dummy byte per receive byte is required). 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 received):
  - The dummy 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. (See Section 31.7, "SPI Interrupts / DMA Requests"). Unlike the command and address phases, the data now contained in SPIRD - SPI RX_Data Register is the 8-bit EEPROM data since the last cycle was initiated to receive data from 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 received.
- The host software will read and store the EEPROM data value in SPIRD - SPI RX_Data Register.

If no more data needs to be received by the master, CS# is released and the SPI is idle. Otherwise, master continues reading the data by writing a dummy value to the TX_DATA register after every 8 SPI_CLK cycles.

Write

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is de-asserted '0' to enable the SPI interface in Full 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.
- Assert WR# high using a GPIO pin.
- Assert CS# low using a GPIO pin.
- Write a valid command word (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. 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.

- Next, EEPROM address A15-A8 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 transmit byte (Command Byte transmitted):
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit command data to the slave. This particular slave device tri-states the SPDIN pin to the master while it is accepting command data. This SPIRD data is ignored.

USER'S NOTE: External pull-up or pull-down is required on the SPDIN pin if it is tri-stated by the slave device.

Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A15-A8) 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. Note: The particular slave device ignores address A15-A13.

- 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, EEPROM address A7-A0 is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, 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 from the second transmit byte (Address Byte (MSB) transmitted):
  - EEPROM address A15-A8 has been transmitted to the slave completing the second SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). 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 second SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A7-A0) 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.
- Next, a data byte (D7:D0) is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, 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 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. (See Section 31.7, "SPI Interrupts / DMA Requests"). 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. (See Section 31.7, "SPI Interrupts / DMA Requests"). 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.

31.12.2 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. (See Section 31.7, "SPI Interrupts / DMA Requests"). 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.
- 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.

//Transmit next reading command

- BIOEN is asserted '1' to indicate that data will now be driven by the master.
- Next, a command byte is written to the TX_DATA register. This value is the first half of a 16 bit command to be sent to temperature sensor peripheral. The SPI master automatically clears the TXFE bit, but does not begin shifting the command data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty. This data will be transmitted because the output buffer is enabled. Data on the SPDIN pin is sampled on each clock.
- After 8 SPI_CLK pulses from the second receive byte:
  - The second SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is the second 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 (command byte 1) 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.

- 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, the second command byte is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the command 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 transmit byte:
  - The third SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 31.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid, since this command was used to transmit the first command byte to the SPI slave.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (command 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.
- Once the TXFE bit is asserted the SPI Master is ready to transmit or 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.
- Since no more data needs to be transmitted, the host software will wait for the RXBF status bit to be asserted indicating the second command byte was transmitted successfully.
- #CS is de-asserted.

32.0 VBAT-POWERED CONTROL INTERFACE

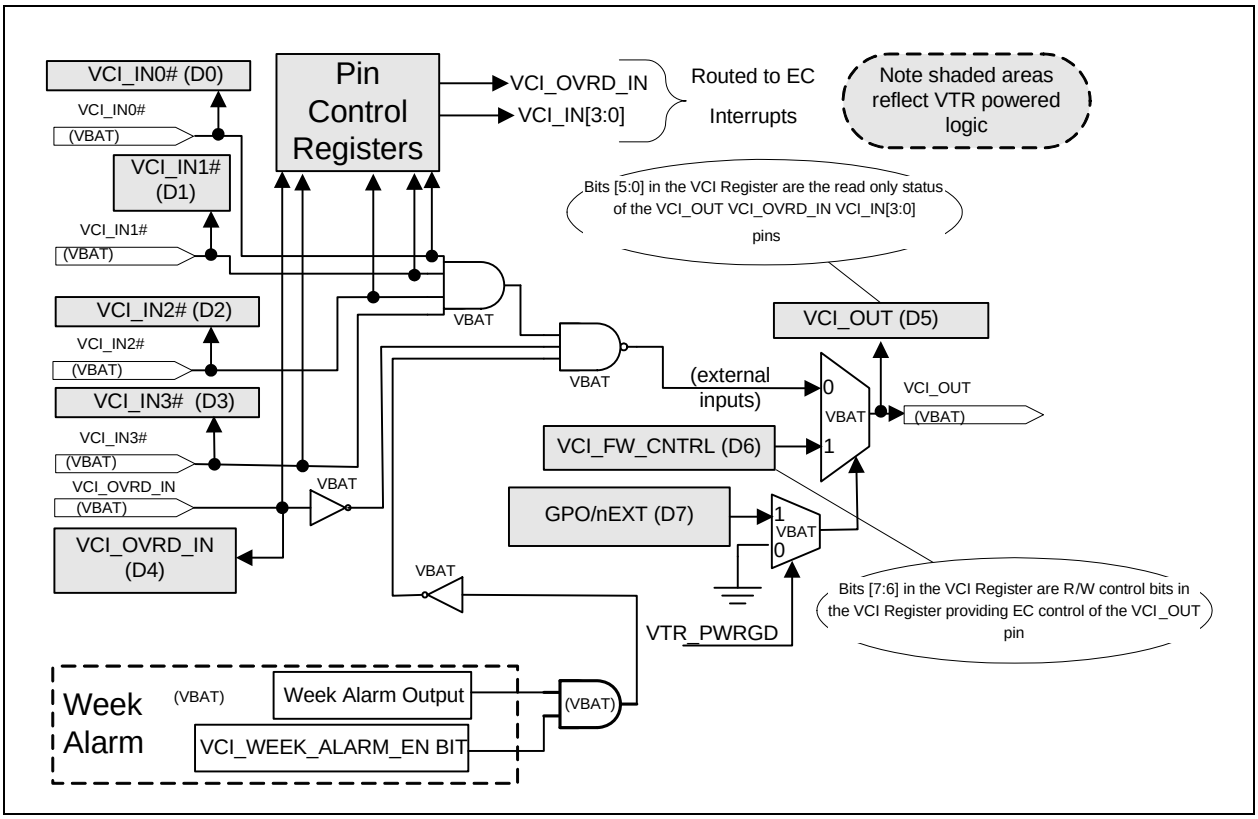
32.1 General Description

The VBAT-Powered Control Interface has VBAT powered combinational logic and Input and output Signal Pins. The VBAT-Powered Control Interface has one VTR powered register. The VBAT-Powered Control Interface block interfaces with the VBAT-powered Week Alarm Interface on page 324 and the VTR-powered Week Timer Control Register on page 326.

32.2 Features

32.3 Block Diagram

FIGURE 32-1: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



Note: Shaded bits are located in this block in the VCI register, all other logic is in the Week Alarm Interface.

32.4 Block Diagram Signal List

Table 2-23, “VCI Interface,” on page 21 includes the Pin Signal Function descriptions for the VBAT-Powered Control Interface.

TABLE 32-1: VBAT-POWERED CONTROL INTERFACE SIGNAL LIST

Signal Name	Direction	Description
VCI_OUT	Output	VBAT-Powered Control Interface Pin Status Bit
VCI_IN[3:0]#	Input	Active low control inputs (see Section 32.8, "Input Timing," on page 440)
VCI_OVRD_IN	Output	Mux control in the VBAT control logic (see Week Alarm Interface)
Week Alarm Output	Input	Asserted signal enables Week Alarm terminal count to assert VCI_OUT
VCI_WEEK_ALARM_EN	Input	Asserted signal enables Week Alarm terminal count to assert VCI_OUT.
Interrupts	Output	See Section 32.6, "Interrupts," on page 438
uP Interface	I/O Bus	Bus used by microprocessor to access the registers in this block.

This block is powered by the VBAT power supply.

32.5 Power, Clocks and Reset

32.5.1 POWER DOMAIN

This block is in the VTR power domain for EC interaction and uses the VBAT power domain for memory retention.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

32.5.2 CLOCKS

The VBAT-Powered Control Interface has one clock input, the EC Bus Clock, which is used to interface to the embedded controller accessible register.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

32.5.3 POWER ON RESET

The VBAT-Powered Control Interface is reset on a VBAT_POR. VCI Register on page 441 are VTR powered and on a nSYS_RST. See Section 5.6.4, "nSYS_RST," on page 98.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

32.6 Interrupts

The VBAT-Powered Control Interface generates interrupts and wakeup events for the following pin inputs:VCI_IN[3:0]# & VCI_OVRD_IN. The Interrupts is routed VCI_OVRD_IN, VCI_IN0, VCI_IN1, VCI_IN2, VCI_IN3, bits in GIRQ23 Source Register. The VBAT-Powered Control Interface Interrupt and wake events can only be asserted when VBAT and VTR is both applied; edge detection is disabled when VTR is unpowered. The edge detection for the interrupt and wake events are controlled each pin by the pin’s associated Pin Control Register. See Pin Control Register on page 337.

32.7 General Description

The MEC1609/MEC1609i VBAT-Powered Control Interface (VCI) is illustrated in Figure 32-1. This block contains the VCI Register. This register monitors the status of the VCI_IN[3:0]#, VCI_OVRD_IN and VCI_OUT pins and also provides firmware control of VCI_OUT when VTR is present.

The state of the VCI_OUT pin can be determined using the Table 32-2, “VCI Output Truth Table,” on page 439. Signals in Table 32-2 are described in Table 32-1, “VBAT-Powered Control Interface Signal List,” on page 438 and in Section 32.10.1, "VCI Register," on page 441.

TABLE 32-2: VCI OUTPUT TRUTH TABLE

Inputs										Output	Description
VCI_OVRD_IN Pin	VCI_IN0# Pin	VCI_IN1# Pin	VCI_IN2# Pin	VCI_IN3# Pin	Week Alarm Output	VCI_WEEK_ALARM_EN	VCI_FW_CNTRL Bit	GPO/nEXT Bit	VTRGD	VCI_OUT Pin	
X	0	X	X	X	X	X	X	X	0	1	VTR = OFF External inputs can drive VCI_OUT
X	X	0	X	X	X	X	X	X	0	1	
X	X	X	0	X	X	X	X	X	0	1	
X	X	X	X	0	X	X	X	X	0	1	
1	X	X	X	X	X	X	X	X	0	1	
X	X	X	X	X	1	1	X	X	0	1	
0	1	1	1	1	0	0	X	X	0	0	
X	0	X	X	X	X	X	X	0	1	1	VTR = ON External inputs can drive VCI_OUT (GPO/nEXT = ‘0’)
X	X	0	X	X	X	X	X	0	1	1	
X	X	X	0	X	X	X	X	0	1	1	
X	X	X	X	0	X	X	X	0	1	1	
X	X	X	X	X	1	1	X	0	1	1	
1	X	X	X	X	X	X	X	0	1	1	
0	1	1	1	1	0	0	X	0	1	0	
X	X	X	X	X	X	X	0	1	1	0	VTR = ON EC drives VCI_OUT (GPO/nEXT = ‘1’)
X	X	X	X	X	X	X	1	1	1	1	

Note: When VTRGD is not asserted ‘0,’ VTR is unpowered and both the EC and the VCI register are unavailable. When VTRGD is asserted ‘1,’ the VTR well is powered, the VCI register is powered, accessible to the EC, and contributes to the resultant logic driving the VCI_OUT pin.

APPLICATION NOTE: The VBAT-Powered Control Interface can be used in a system as follows:

1. The initial condition is the VBAT battery is installed causing a VBAT_POR, no AC power applied and no power button event. Therefore no power is applied to VTR power well. The VCI_OUT pin is de-asserted, the EC is not running, the Week Alarm has not been initialized or enabled.
2. Either applying AC charger power and asserting the VCI_OVRD_IN input or depressing a power button asserting a one or more of the VCI_IN[3:0]# input pins causes the VCI_OUT pin to be asserted (maybe a pulse for seconds). This powers up the VTR power well. This starts the EC and allows access to the VCI register.
3. If the VCI register is not written then the VBAT-Powered Control Interface glue logic behaves the same way as when VTR is empowered; the VCI_OUT pin is asserted when any one of the an external input (FIGURE 32-1: on page 437) are asserted: the VCI_OVRD_IN, or VCI_IN[3:0]#, or the Week Alarm Power-up signal are asserted.
4. The EC can read the status of the VCI_OVRD_IN and VCI_IN[3:0]# in the VCI register (similar to a GP input.) The EC can enable interrupts from these pins.
5. The EC can take programmable control of the VCI_OUT pin output by setting both the VCI_FW_CNTRL and the GPO/nEXT bits in the VCI register.

Note: BIOS should set VCI_FW_CNTRL bit to 1 prior to setting the GPO/nEXT bit to 1 to ensure a glitch free VCI_OUT pin output.

6. Clearing the VCI_FW_CNTRL bit and setting the GPO/nEXT bit in the VCI register causes the VCI_OUT pin to be de-asserted. The VCI_OUT pin remains de-asserted until VTR is empowered. When VTR=0, the VCI_OVRD_IN and VCI_IN[3:0]# inputs pins control the VCI_OUT pin.

Note:	The VCI_IN[3:0]# pins have no direct effect on the VCI_OUT pin when VCI_OVRD_IN is asserted. However, when VTR is on and EC is running, the VCI_IN[3:0]# pins can cause an interrupts and the EC can de-assert the VCI_OUT pin by programming the VCI_FW_CNTRL and the GPO/nEXT bits in the VCI register. The VCI_OVRD_IN and the VCI_IN[3:0]# pins can also cause EC wake events.

The state of the VCI_OUT pin can be determined using the Table 32-2, “VCI Output Truth Table,” on page 439. Signals in Table 32-2 are described in Table 32-1, “VBAT-Powered Control Interface Signal List,” on page 438 and in Section 32.10.1, "VCI Register," on page 441.

Note:	When VTRGD on page 96 is not asserted ‘0,’ VTR is unpowered and both the EC and the VCI register are unavailable. When VTRGD is asserted ‘1,’ the VTR well is powered, the VCI register is powered, accessible to the EC, and contributes to the resultant logic driving the VCI_OUT pin.

32.8 Input Timing

When VTRGD is not asserted, or VTRGD is asserted and the GPO/nEXT bit is ‘0,’ transition times on the VCI_IN[3:0]# and VCI_OVRD_IN input pins must not be greater than 1 µs (Figure 32-2, Table 32-3).

FIGURE 32-2: VBAT-Powered Control Interface Input Timing

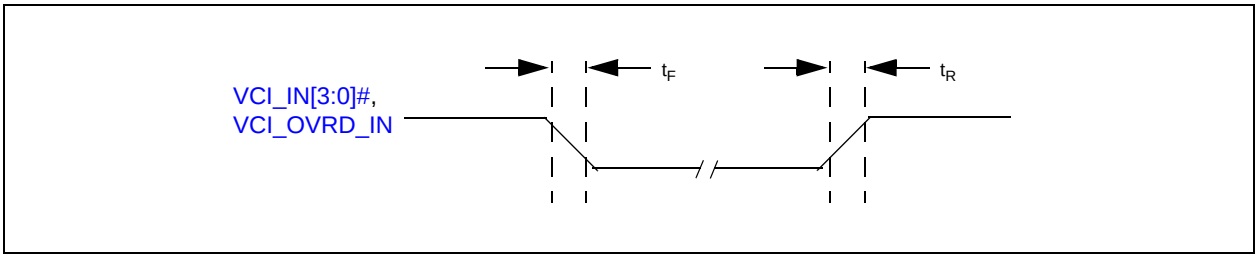


TABLE 32-3: INPUT TIMING PARAMETERS

Parameter	Symbol	MIN	TYP	MAX	Unit	Conditions
VCI_IN[3:0]# and VCI_OVRD_IN Rise and Fall Time	t_R, t_F	–	–	1	µs	VTRGD = ‘0,’ or VTRGD = ‘1’ and GPO/nEXT = ‘0.’

32.9 Registers

The VBAT-Powered Control Interface has its own Logical Device Number, and Base Address as indicated in Table 32-4.

Table 32-5 is a register summary for the VBAT-Powered Control Interface block.

Each instance of the VBAT-Powered Control Interface Base Address as indicated in Table 32-4.

TABLE 32-4: VBAT-Powered Control Interface BASE ADDRESS TABLE

VBAT-Powered Control Interface **Instance**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
VBAT-Powered Control Interface	3Fh	FF_FD00h = FF_FC00h+100h

Note:	The VBAT-Powered Control Interface is in the Global Configuration Logical Device, Host Logical Device Number 3Fh, in EC-only space (see Section 3.4.2, "Address Framing," on page 51).

Table 32-5 is a register summary for the VBAT-Powered Control Interface block. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 32-5: VBAT-POWERED CONTROL INTERFACE REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
VCI register	00h	0	R	

32.10 Detailed Description of Accessible Registers

32.10.1 VCI REGISTER

The VCI Register contains various pin status indicators and control bits.

TABLE 32-6: VCI REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	00					8-bit	**EC SIZE**	
POWER	VTR					0000xxxxb	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R	R	R	R	R	R
BIT NAME	GPO/ NEXT	VCI_ FW_ CNTRL	VCI_ OUT	VCI_ OVRD_ IN	VCI_IN[3:0]#			

Note: The VCI_IN[3:0]#, VCI_OVRD_IN, & VCI_OUT pins bits default at VCC1_POR to the current state of their respective input pins.

VCI_IN[3:0]#

These read-only bits provide the current status of the associated VCI_IN[3:0]# pins.

VCI_OVRD_IN

This read-only bit provides the current status of the VCI_OVRD_IN pin.

VCI_OUT

This read-only bit provides the current status of the VCI_OUT pin.

VCI_FW_CNTRL

The VCI_FW_CNTRL allows EC firmware to control the VCI_OUT pin. Clearing this bit can de-assert the active high VCI_OUT.

APPLICATION NOTE: BIOS should set VCI_FW_CNTRL bit to 1 prior to setting the GPO/nEXT bit to 1 on power up.

GPO/NEXT

The GPO/nEXT bit controls the Mux selecting between the external VBAT-Powered Control Interface Signal List inputs and the VCI_FW_CNTRL bit output. When GPO/nEXT is set (‘1’), VCI_OUT follows the VCI_FW_CNTRL bit setting. When GPO/nEXT is clear (‘0’), VCI_OUT follows the external inputs.

33.0 VBAT POWERED RAM

33.1 Abstract

33.1.1 CHIP LEVEL INTERFACE CLOCK DOMAIN/POWER DOMAIN CROSSINGS

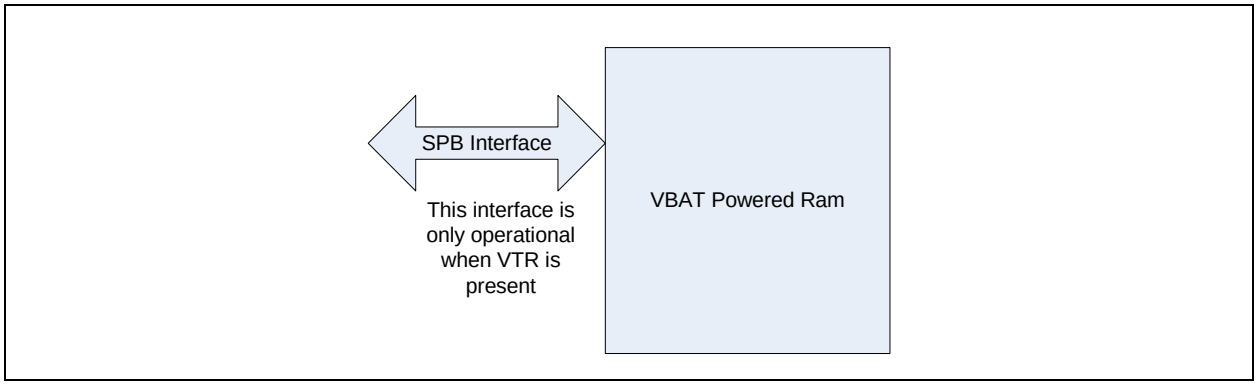
Powered by VBAT but only clocked or accessed when VTR power is available.

33.2 General Description

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

33.3 Block Diagram

FIGURE 33-1: BLOCK DIAGRAM OF VBAT POWERED RAM



33.4 Power, Clocks and Reset

33.4.1 POWER DOMAIN

This block is in the VTR power domain for EC interaction and uses the VBAT power domain for memory retention.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

33.4.2 CLOCKS

The VBAT Powered RAM has one clock input., the EC Bus Clock.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

33.4.3 POWER ON RESET

The VBAT Powered RAM is reset on a VBAT_POR.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

33.5 Interrupts

The VBAT Powered RAM has not interrupts.

33.6 Registers

The VBAT Powered RAM has it's own Logical Device Number, and Base Address as indicated in Table 33-1. See Note 3-1 on page 49.

TABLE 33-1: VBAT Powered RAM BASE ADDRESS TABLE

VBAT Powered RAM Blocks	LDN from (Table 3-2 on page 48)	AHB Base Address
VBAT Backed Memory	33h	F0_CD00h

Each 32-bit RAM location is an SPB Offset from the AHB base address.

34.0 LED INTERFACE

34.1 General Description

The LED Interface can control three external LEDs. Each LED can be individually set to be full on, full off, or oscillate. Oscillation can in turn be configured to “blink”, where the LED output switches between full on and full off at a fixed frequency, or to “breathe”, where the brightness of the LED increases and decreases at a fixed rate. The periodic behavior of the LEDs is driven by the 32.768KHz crystal oscillator clock, so the LED Interface continues to function even when the EC is in sleep mode or the 64.52 MHz Ring Oscillator is shut down.

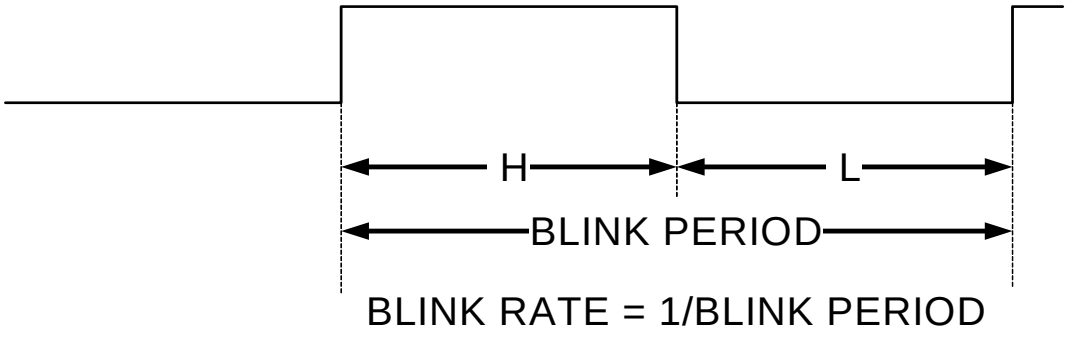
The Blink Mode equations are shown in Figure 34-1 and Breathing Mode LED Equations are shown in Figure 34-2.

FIGURE 34-1: BLINK MODE EQUATIONS

$$\text{BLINK RATE} = \frac{32768}{(\text{LED\_PRESCALE[11:0]} + 1)} \times \frac{1}{2^8} \text{ Hz}$$

$$\text{“H” width} = \frac{\text{LED\_DUTYCYCLE[7:0]}}{\text{BLINK RATE}} \times \frac{1}{256} \text{ Sec}$$

$$\text{“L” width} = \frac{(256 - \text{LED\_DUTYCYCLE[7:0]})}{\text{BLINK RATE}} \times \frac{1}{256} \text{ Sec}$$



BLINK RATE = 1/BLINK PERIOD

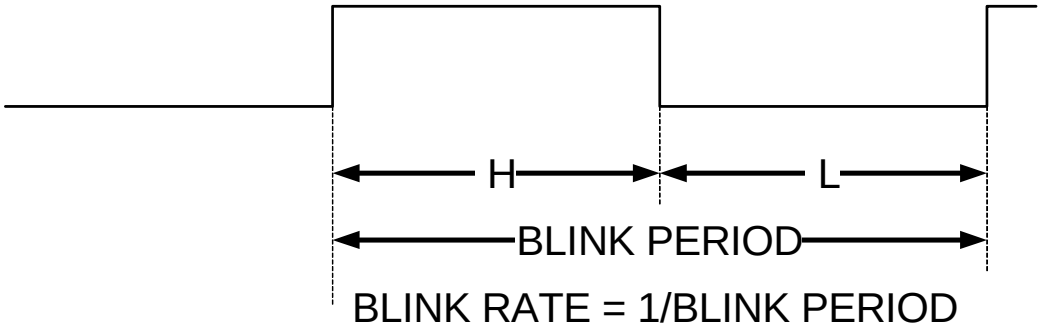
FIGURE 34-2: BREATHING MODE LED EQUATIONS

$$\text{BLINK RATE} = \frac{32768}{(\text{LED\_PRESCALE[11:0]} + 1)} \times \frac{1}{2^8} \text{ Hz}$$

$$\text{“H” width} = \frac{\text{LED\_DUTYCYCLE[7:0]}}{\text{BLINK RATE}} \times \frac{1}{256} \text{ Sec}$$

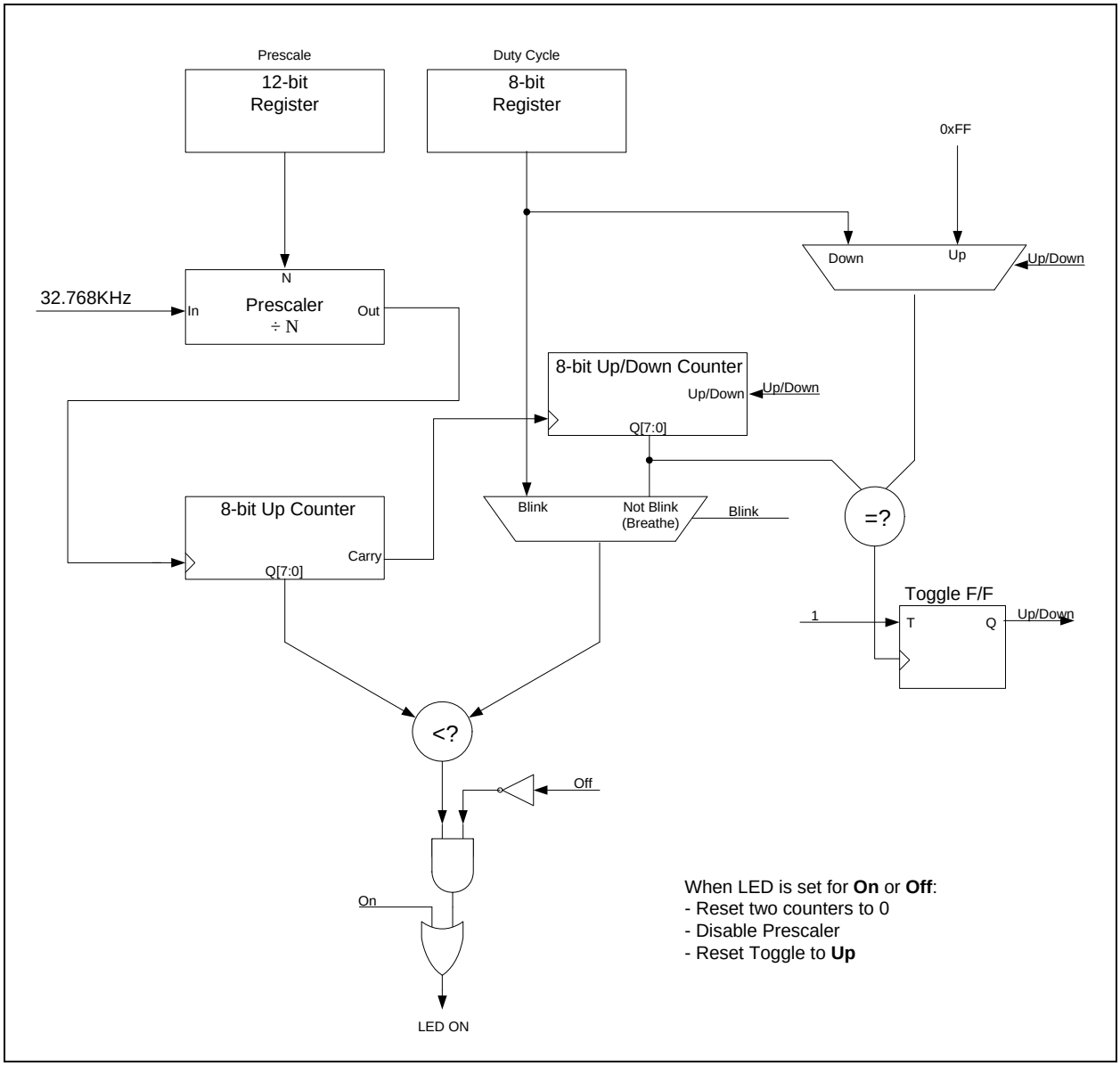
$$\text{“L” width} = \frac{(256 - \text{LED\_DUTYCYCLE[7:0]})}{\text{BLINK RATE}} \times \frac{1}{256}$$

$$\text{Breathing Oscillation TIme} = \frac{2x\ (255 - \text{“Min” LED\_DUTYCYCLE[7:0]})}{\text{BLINK RATE}} \text{ Sec}$$



34.2 LED Block Diagram

FIGURE 34-3: LED BLOCK DIAGRAM



34.3 Block Diagram Signal List

TABLE 34-1: LED Interface PORT LIST

Signal Name	Direction	Description
32.768KHz	INPUT	X32K_CLK, 32.768 crystal controlled clock
Blink	Internal	Control signal from LED Control Register
Up/Down	Internal	Control signal generated by 8-bit Up/Down counter
LED ON	OUTPUT	LED outputs

34.4 Power, Clocks and Reset

34.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

34.4.2 CLOCKS

This block has two clock inputs, the EC Bus Clock and X32K_CLK. The EC Bus Clock is used in the interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK is the clock source for the LED functional logic, including the counters.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

34.4.3 RESET

This block is reset on a nSYS_RST.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

34.5 Interrupts

The LED block does not generate interrupts.

34.6 LED Blinking and Breathing

Blinking and breathing is controlled by two registers for each LED. The first register controls the clock prescaler that sets the oscillation period. An 8-bit counter clocked on the pre-scaled 32.768KHz clock defines a blink period with 256 phases. In “blink” mode, the second register determines the duty cycle of the LED blink. In “breathe” mode, the second register determines the minimum duty cycle of the LED.

When the prescale is 0, the blink period will use the 32.768KHz clock (with 30.5µs phases). For $N>0$, the 32.768KHz clock will be divided by $N+1$. For examples of settings of the prescale and duty cycle registers, see Table 34-2, "LED Control Configuration Examples". The maximum blink period is 32 seconds.

When an LED is configured to be fully off or fully on, the prescalar and other counters in the LED circuitry are shut down in order to save power.

34.6.1 BLINKING

When configured for blinking, the LED will be on for all phases of the prescaled period that are less than the duty cycle and off for all phases that are greater than the duty cycle. 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.

34.6.2 BREATHING

When configured for breathing, the duty cycle of the LED blink will continuously increase and decrease between full on (a duty cycle of FFh) and a minimum duty cycle set by the duty cycle register. After each blink period the duty cycle will increase by 1, until the duty cycle saturates at FFh. Once the duty cycle saturates, it is reduced by 1 after each blink period, until it reaches a minimum duty cycle set by the duty cycle register. Once the minimum duty cycle is reached, the duty cycle will start increasing again. If the frequency of the LED blink period is sufficiently fast (for example, greater than 30Hz), the LED will not appear to blink but will instead appear dimmer or brighter, depending on the duty cycle.

The overall duration of the breathing oscillation is a factor of the blink period and the minimum duty cycle. The total time will be 2 X (*BLINK_PERIOD* X (*FFh - MIN_DUTY_CYCLE*)).

TABLE 34-2: LED CONTROL CONFIGURATION EXAMPLES

Prescale	Duty Cycle	Blink Rate	Blink	Breathe
000h	00h	128Hz	full off	full off to full on 4s oscillation cycle
001h	40h	64Hz	3.9ms on, 11.6ms off	quarter on to full on 6s oscillation cycle
003h	80h	32Hz	15.5ms on, 15.5ms off	half on to full on 8s oscillation cycle

TABLE 34-2: LED CONTROL CONFIGURATION EXAMPLES (CONTINUED)

Prescale	Duty Cycle	Blink Rate	Blink	Breathe
07Fh	20h	1Hz	125ms on, 0.875s off	blink to 1s on 7m 26s oscillation cycle
0BFh	16h	0.66Hz	125ms on, 1.375s off	blink to 1.5s on 11m 39s oscillation period
0FFh	10	0.5Hz	125ms on, 1.875s off	blink to 2s on 15m 56s oscillation period
180h	0Bh	0.33Hz	125ms on, 2.875s off	blink to 3s on 24m 24s oscillation period
1FFh	40h	0.25Hz	1s on, 3s off	1s/3s on/off to 4s on 12m 48s oscillation cycle

34.7 LED Registers

There are three instances of the LED Interface block implemented in the MEC1609/MEC1609i enumerated as [2:0]. Each instance of the LED Interface has its Base Address as indicated in Table 34-3, "LED Interface Base Address Table".

TABLE 34-3: LED Interface BASE ADDRESS TABLE

LED **Instance**	**LDN from (Table 3-2 on page 48)**	**AHB Base Address**
LED.**0**	21h	F0_8400h
LED.**1**		F0_8400h + 80h
LED.**2**		F0_8400h + 100h

Table 34-4 is a register summary for one instance of the LED Interface block.

TABLE 34-4: LED REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
LED Control Register	00h	0	R/W	
LED Rate Register	04h	3-0	R/W	

Note: It may take up to 30µs for a change to an LED register to take effect.

34.7.1 LED CONTROL REGISTER

The LED Control Register is used to control the behavior of each of the three output LEDs.

TABLE 34-5: LED CONTROL REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	0h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved					Synch	Control	

CONTROL

This bit controls the behavior of LED:

0= LED is always off

1= LED blinks, at a rate controlled by the LED Rate Register

2= LED breathes, at a rate controlled by the LED Rate Register

3= LED is always on

SYNCH

When this bit is 1, all counters for all LEDs are reset to their initial values. When this bit is 0 in the LED Control Register for all LEDs, then all counters for LEDs that are configured to blink or breathe will increment or decrement, as required.

APPLICATION NOTE: To synchronize blinking or breathing, the Synch bit should be set for at least one LED, the LED Control Register and the LED Rate Register for each LED should be set to their required values, then the Synch bits should all be cleared. If the LED Rate Registers are set for the same blink period, they will all be synchronized.

34.7.2 LED RATE REGISTER

The LED Rate Register is used to configure the blinking and breathing rate of each of the LEDs.

TABLE 34-6: LED RATE REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	04h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				LED_Prescale[11:8]			
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	LED_Prescale[7:0]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	LED_DutyCycle							

LED_DUTYCYCLE[7:0]

The field determines the duty cycle of the LED blink pattern. A value of 0 means full off, a value of FFh means full on.

LED_PRESCALE[11:0]

If this field is 0, the 32.768KHz clock will be used to determine the blink period of LED: If this field is greater than 0, then the 32.768KHz clock will be divided by LED_Prescale[11:0]+1.

35.0 PS/2 DEVICE INTERFACE

35.1 General Description

There are three PS/2 Ports in the MEC1609/MEC1609i independent EC PS/2 serial ports implemented in hardware which are directly controlled by the EC (see FIGURE 35-1: on page 451). The hardware implementation eliminates the need to bit bang I/O ports to generate PS/2 traffic, however bit banging is available via the associated GPIO's.

Each EC PS/2 serial channels use a synchronous serial protocol to communicate with the auxiliary device. Each PS/2 channel has Clock and Data signal lines. The signal lines are bi-directional and employ open drain outputs capable of sinking 16mA. A pull-up resistor, typically 10K, is connected to both lines. This allows either the MEC1609/MEC1609i EC PS/2 logic or the auxiliary device to drive the lines. Regardless of the drive source, the auxiliary device always provides the clock for transmit and receive operations. The serial packet is made up of eleven bits, listed in the order they appear on the data line: start bit, eight data bits (least significant bit first), odd parity, and stop bit. Each bit cell is from 60µS to 100µS long.

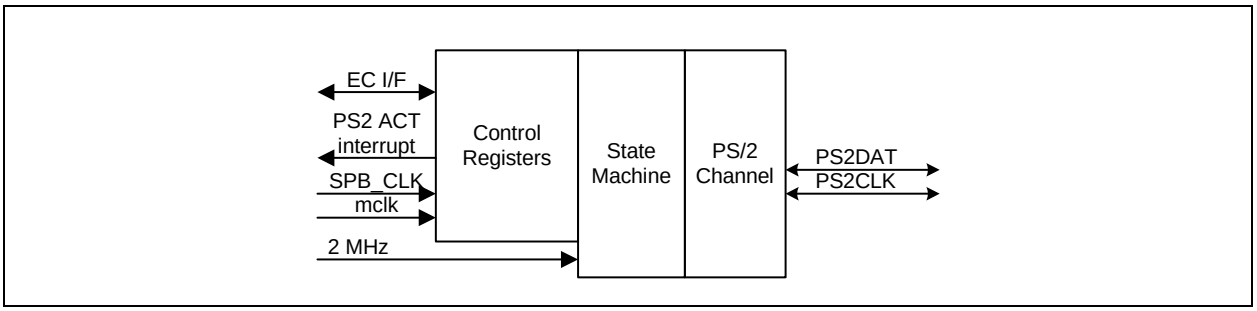
All PS/2 Serial Channel signals (CLK and DAT) are driven by open drain drivers which can be pulled to VTR or VCC (+3.3V nominal) through 10K-ohm resistors.

The MEC1609/MEC1609i supports a PS/2 Wake Interface that can wake the EC from the IDLE or SLEEP states. The PS/2 Wake Interface is powered by VTR and can generate wake interrupts without a clock.

APPLICATION NOTE: the PS/2 Wake Interface is only active when the KBC/Mouse and external pull-up resisters are powered by the VCC1supply. The external pull-up resister must always be powered by the same source as the KBC/Mouse.

35.2 Block Diagram

FIGURE 35-1: PORT PS/2 BLOCK DIAGRAM



35.3 PS/2 Port Physical Layer Byte Transmission Protocol

The PS/2 physical layer transfers a byte of data via an eleven bit serial stream as shown in Table 35-1. A logic 1 is sent at an active high level. Data sent from a Keyboard or mouse device to the host is read on the falling edge of the clock signal. The Keyboard or mouse device always generates the signal. The Host may inhibit communication by pulling the Clock line low. The Clock line must be continuously high for at least 50 microseconds before the Keyboard or mouse device can begin to transmit its data. See Table 35-2, "PS/2 Port Physical Layer Bus States".

TABLE 35-1: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

Bit	Function
1	Start bit (always 0)
2	Data bit 0 (least significant bit)
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4

TABLE 35-1: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL (CONTINUED)

Bit	Function
7	Data bit 5
8	Data bit 6
9	Data bit 7 (most significant bit)
10	Parity bit (odd parity)
11	Stop Bit (always 1)

FIGURE 35-2: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

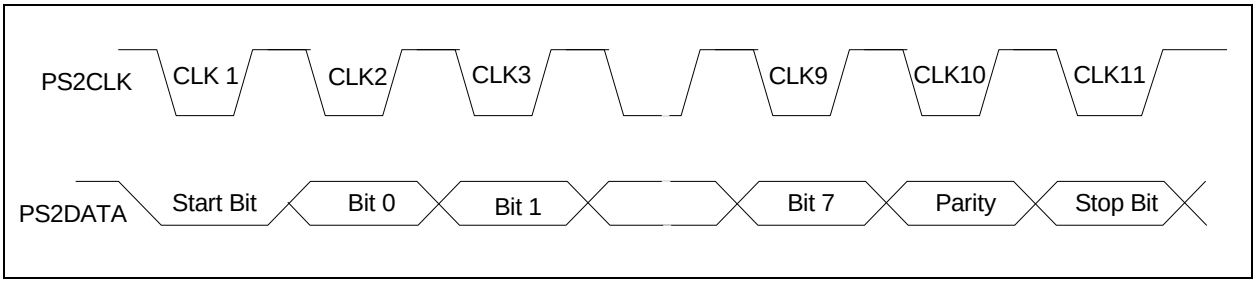


TABLE 35-2: PS/2 PORT PHYSICAL LAYER BUS STATES

Data	Clock	State
high	high	Idle
high	low	Communication Inhibited
low	low	Request to Send

35.4 Block Diagram Signal List

TABLE 35-3: PS/2 PORT LIST

Signal Name	Direction	Description
PS2DAT	INPUT/OUTPUT	Data from the PS/2 device
PS2CLK	INPUT/OUTPUT	Clock from the PS/2 device
nSYS_RST	INPUT	Chip Power on Reset (i.e., Suspend Well)
SPB_CLK	INPUT	Clock Source to EC micro controller. Used for reading/writing registers on the EC memory i/f.
mclk	INPUT	MCLK
2 MHz	INPUT	MCLK_DIV32_EN, State machine clock
SLEEP_EN	INPUT	External Power Management enable/disable input signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode.
CLK_REQ	OUTPUT	Power Management output indicates when this block requires MCLK input. 0= Clock can be turned 'off' when appropriate 1= Clock is required to be 'on.'
EC Interface	I/O BUS	EC-side SPB bus
PS2 ACT	OUTPUT	Asynchronous Interrupt

35.5 Power, Clocks and Reset

35.5.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

35.5.2 CLOCKS

This block uses the EC Bus Clock, the 64.52MHz MCLK and the 2MHz MCLK_DIV32_EN. EC Bus Clock is used when reading and writing the PS/2 Device Interface registers. The state machine is clocked with MCLK_DIV32_EN.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

35.6 Reset

This block is reset on a nSYS_RST.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

35.7 Instance Description

There are three block instances defined in this chapter: PS/2[0,1,2]. The pin signals are defined in Table 2-16, “PS/2 Interface,” on page 18. PS/2 ports ending with signal functions ending with “A” or “B” are muxed to a single controller. Only one Port set of clock and data are intended to used at a time (either “A” or “B” not both.) The unused port segment should have its associated Pin Control Register’ s, Mux Control Field programmed away from the PS2 controller. See Section 22.0, "GPIO Interface," on page 329.

35.8 Interrupts

Each EC PS/2 Channel has two interrupts: a PS/2 activity interrupt event and a START Bit detection Wake-up event. Each PS/2 Channel activity interrupt event is generated by changes in status bits in this block. The PS/2 Channel activity interrupt event are routed to the PS2_ACT_0, PS2_ACT_1,PS2_ACT_2 bits in the GIRQ19 Source Register on page 281. The START Bit detection Wake-up event is a PS/2 Channel/segment (see Section 35.9) Data pin signal edge detection interrupt and wake event. Each PS/2 Channel/segment START Bit detection Wake-up event is controlled by their associated Data pin’s Pin Control Register. (See Section 22.0, "GPIO Interface," on page 329. The START Bit detection Wake-up events routed to the PS2_WK_0A, PS2_WK_0BPS2_WK_1A, PS2_WK_1B, PS2_WK_2 bits in the GIRQ19 Source Register on page 281.

APPLICATION NOTE: The pin control registers for a PS2 wakeup event should be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

35.9 Registers

There are three block instances defined in this chapter: PS/2[0,1,2]. The pin signals are defined in Table 2-16, “PS/2 Interface,” on page 18. PS/2 ports ending with signal functions ending with “A” or “B” are muxed to a single controller. Only one Port set of clock and data are intended to be used at a time (either “A” or “B” not both.) The unused port segment should have its associated Pin Control Register’s Mux Control Field programmed away from the PS2 controller. (See Section 22.0, "GPIO Interface," on page 329).

Each instance of the PS/2 Device Interface has its Base Address as indicated in Table 35-4.

TABLE 35-4: PS/2 Device Interface BASE ADDRESS TABLE

PS/2 Device Interface **Instance**	**LDN from (**Table 3-2 on page 48**)**	**AHB Base Address**
PS/2.0	22h	F0_8800h
PS/2.1		F0_8880h = F0_8800h + 80h
PS/2.2		F0_8900h = F0_8800h + 100h

Table 35-5 is a register summary for each instance of the PS/2 Device Interface.

TABLE 35-5: PS/2 Device Interface REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
PS/2 Transmit Buffer Register	00h	0	W	
PS/2 Receive Buffer Register	00h	0	R	
PS/2 Control Register	04h	0	R/W	
PS/2 Status Register	08h	0	R/WC	

35.10 Detailed Description of Accessible Registers

35.10.1 PS/2 TRANSMIT BUFFER

TABLE 35-6: PS/2 TRANSMIT BUFFER REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	00h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	--	-	-	-	-
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	Transmit Data[7:0]							

TRANSMIT DATA

The byte written to this register, when PS2_T/R and PS2_EN in the PS/2 Control Register and XMIT_IDLE in the PS/2 Status Register are set, is transmitted automatically by the PS/2 channel control logic. If any of these three bits (PS2_T/R, PS2_EN, and XMIT_IDLE) are not set, then writes to this register are ignored. On successful completion of this transmission or upon a Transmit Time-out condition, the PS2_T/R bit is automatically cleared and the XMIT_IDLE bit is automatically set. The PS2_T/R bit must be written to a '1' before initiating another transmission to the remote device.

Note:

- Even if PS2_T/R, PS2_EN, and XMIT_IDLE are all set, writing the Transmit Register will not start a transmission if RDATA_RDY in the PS/2 Status Register is set. The automatic PS/2 logic forces data to be read from the Receive Register before allowing a transmission.
- An interrupt is generated on the low to high transition of XMIT_IDLE.
- All bits of this register are write only.

35.10.2 PS/2 RECEIVE BUFFER

TABLE 35-7: PS/2 RECEIVE BUFFER REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	00h					8-bit	**EC SIZE**	
POWER	VTR					FFh	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	--	-	-	-	-
EC TYPE	R							
BIT NAME	Receive Data[7:0]							

RECEIVE DATA

When PS2_EN=1 and PS2_T/R=0 in the PS/2 Control Register, the PS2 Channel is configured to automatically receive data on that channel (both the CLK and DATA lines will float waiting for the peripheral to initiate a reception by sending a start bit followed by the data bits). After a successful reception, data is placed in this register and the RDATA_RDY bit in the PS/2 Status Register is set and the CLK line is forced low by the PS2 channel logic. RDATA_RDY is cleared and the CLK line is released to high-z following a read of this register. This automatically holds off further receive transfers until the EC has had a chance to get the data.

Note:

- The Receive Register is initialized to FFh after a read or after a Time-out has occurred.
- The channel can be enabled to automatically transmit data (PS2_EN=1) by setting PS2_T/R while RDATA_RDY is set, however a transmission can not be kicked off until the data has been read from the Receive Register.
- An interrupt is generated on the low to high transition of RDATA_RDY.
- If a receive time-out (REC_TIMEOUT=1 in the PS/2 Control Register) or a transmit time-out (XMIT_TIMEOUT=1 in the PS/2 Control Register) occurs the channel is busied (CLK held low) for 300us (Hold Time) to ensure that the peripheral aborts. Writing to the Transmit Register will be allowed, however the data written will not be transmitted until the Hold Time expires.
- All bits in this register are read only

Note 35-1 In receive mode the RX_BUSY bit for a particular channel is set in the PS/2 Status Register.

35.10.3 PS/2 CONTROL

There are three PS/2 Control Registers, one for each channel.

TABLE 35-8: PS/2 CONTROL REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	04h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		STOP		PARITY		PS2_EN	PS2_T/R

PS2_T/R

PS/2 Channel Transmit/Receive (default = 0). Configures the PS2 logic for automatic transmission when set or reception when cleared. This bit is only valid when PS2_EN is set.

When set the PS/2 channel is enabled to transmit data. To properly initiate a transmit operation, this bit must be set prior to writing to the Transmit Register. Writes to the Transmit Register are blocked when this bit is cleared. Upon setting the PS2_T/R bit, the channel will drive its CLK line low and then float the DATA line and hold this state until a write occurs to the Transmit Register or until the PS2_T/R bit is cleared. Writing to the Transmit Register initiates the transmit operation. MEC1609/MEC1609i drives the data line low and, within 80ns, floats the clock line (externally pulled high by the pull-up resistor) to signal to the external PS/2 device that data is now available. The PS2_T/R bit is cleared on the 11th clock edge of the transmission or if a Transmit Time-out error condition occurs.

Note:	If the PS2_T/R bit is set while the channel is actively receiving data prior to the leading edge of the 10th (parity bit) clock edge, the receive data is discarded. If this bit is not set prior to the 10th clock signal, then the receive data is saved in the Receive Register.

When the PS2_T/R bit is cleared, the PS/2 channel is enabled to receive data. Upon clearing this bit, if RDATA_RDY is also cleared, the channel's CLK and DATA will float waiting for the external PS/2 device to signal the start of a transmission. If the PS2_T/R bit is set while RDATA_RDY is set, then the channel's DATA line will float but its CLK line will be held low, holding off the peripheral, until the Receive Register is read.

PS2_EN

PS2 Channel ENable (default = 0). When PS2_EN is set, the PS/2 State machine is enabled allowing the channel to perform automatic reception or transmission depending on the bit value of PS2_T/R. When PS2_EN is cleared, the channel's automatic PS/2 state machine is disabled and the PS/2 channel's CLK pin driven low and DATA pin not driven.

Note:	If the PS2_EN bit is cleared prior to the leading edge (falling edge) of the 10th (parity bit) clock edge the receive data is discarded (RDATA_RDY remains low). If the PS2_EN bit is cleared following the leading edge of the 10th clock signal, then the receive data is saved in the Receive Register (RDATA_RDY goes high) assuming no parity error.

PARITY

These bits are used to set the parity expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set.

00=Receiver expects Odd Parity (default).

01=Receiver expects Even Parity.

10=Receiver ignores level of the parity bit (10th bit is not interpreted as a parity bit).

11=Reserved

STOP

These bits are used to set the level of the stop bit expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set.

00=Receiver expects an active high stop bit.

01=Receiver expects an active low stop bit.

10=Receiver ignores the level of the Stop bit (11th bit is not interpreted as a stop bit).

11=Reserved.

APPLICATION NOTE: Changing values in the control register at a rate faster than 2 MHz, may result in unpredictable behavior.

This register should be read to determine the status of PS2_T/R and PS2_EN prior to clearing by writing a 1 to that bit.

35.10.4 PS/2 STATUS

TABLE 35-9: PS/2 STATUS REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	08h					8-bit	**EC SIZE**	
POWER	VTR					10h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R	R/WC	R	R/WC	R/WC	R/WC	R
BIT NAME	XMIT_ START_ TIMEOUT	RX_ BUSY	XMIT_ TIME-OUT	XMIT_ IDLE	FE	PE	REC_ TIME-OUT	RDATA_R DY

APPLICATION NOTE: This register should be read to determine the status of R/WC bits prior to clearing by writing a 1 to that bit.

RDATA_RDY

Receive Data Ready: Under normal operating conditions, this bit is set following the falling edge of the 11th clock given successful reception of a data byte from the PS/2 peripheral (i.e., no parity, framing, or receive time-out errors) and indicates that the received data byte is available to be read from the Receive Register. This bit may also be set in the event that the PS2_EN bit is cleared following the 10th CLK edge. Reading the Receive Register clears this bit.

Note: An Interrupt is generated on the low-to-high transition of the RDATA_RDY bit.

REC_TIMEOUT

Following assertion of the REC_TIMEOUT bit, the channel's CLK line is automatically pulled low for a minimum of 300us until the PS/2 status register is read. Under PS2 automatic operation, PS2_EN is set, this bit is set on one of three receive error conditions:

When the receiver bit time (time between falling edges) exceeds 300us.

If the time from the first bit (start) to the 10th bit (parity) exceeds 2ms.

On a receive parity error along with the Parity Error (PE) bit.

On a receive framing error due to an incorrect STOP bit along with the framing error (FE) bit.

The REC_TIMEOUT bit is cleared when the Status Register is read.

Note: An Interrupt is generated on the low-to-high transition of the REC_TIMEOUT bit.

PE

Parity Error: When receiving data, the parity bit is clocked in on the falling edge of the 10th CLK edge. If the channel is configured to expect either even or odd parity and the 10th bit is contrary to the expected parity, then the PE and REC_-TIMEOUT bits are set following the falling edge of the 10th CLK edge and an interrupt is generated.

FE

Framing Error: When receiving data, the stop bit is clocked in on the falling edge of the 11th CLK edge. If the channel is configured to expect either a high or low stop bit and the 11th bit is contrary to the expected stop polarity, then the FE and REC_TIMEOUT bits are set following the falling edge of the 11th CLK edge and an interrupt is generated.

XMIT_IDLE

Transmitter Idle: When low, the XMIT_IDLE bit is a status bit indicating that the PS/2 channel is actively transmitting data to the PS2 peripheral device. Writing to the Transmit Register when the channel is ready to transmit will cause the XMIT_IDLE bit to clear and remain clear until one of the following conditions occur: the falling edge of the 11th CLK, XMIT_TIMEOUT is set; the PS2_T/R bit is cleared or the PS2_EN bit is cleared.

Note 35-2 An interrupt is generated on the low-to-high transition of XMIT_IDLE.

XMIT_TIMEOUT

When the XMIT_TIMEOUT bit is set, the PS2_T/R bit is held clear, the PS/2 channel's CLK line is pulled low for a minimum of 300us until the PS/2 Status register is read. The XMIT_TIMEOUT bit is set on one of three transmit conditions: when the transmitter bit time (time between falling edges) exceeds 300us, when the transmitter start bit is not received within 25ms from signaling a transmit start event or if the time from the first bit (start) to the 10th bit (parity) exceeds 2ms.

RX_BUSY

When a RX_BUSY bit is set, the associated channel is actively receiving PS/2 data; when a RX_BUSY bit is clear, the channel is idle. See Note 35-1 on page 455.

XMIT_START_TIMEOUT

When the XMIT_START_TIMEOUT bit is set, a start bit was not received within 25 ms following the transmit start event. Writing a '1' to the bit clears the XMIT_START_TIMEOUT bit. The XMIT_START_TIMEOUT bit is a 'sticky' bit and is intended to uniquely indicate the status of the transmit start bit time-out condition. These bit affect no other logic. Note that the transmit start bit time-out condition is also indicated by the XMIT_TIMEOUT bit.

PROGRAMMER'S NOTE: Always check that an EC PS/2 channel is idle, i.e. the RX_BUSY bit is clear, before attempting to transmit on that channel. Receive data may be lost by setting an EC PS/2 channel to transmit while the RX_BUSY bit is set depending where in the message frame the transmit mode change occurs.

35.11 Power Management

TABLE 35-10: PS/2 Device Interface Power Management

PS2_EN Bit	SLEEP_EN	Block Idle Status (Note 35-3)	CLK_REQ	State	Description
0	X	X	0	SLEEPING	The block is disabled and the clock can be stopped.
1	0	NOT IDLE	1	NORMAL OPERATION	The block is not idle and neither disabled by firmware nor commanded to sleep.
	1	NOT IDLE	1	PREPARING TO SLEEP	The block is commanded to sleep, but the clock is required until the Block is idle.
	1	IDLE	0	SLEEPING	The block is commanded to sleep and idle. The clock can be stopped.

Note 35-3 The PS/2 Device Interface 'idle' state is determined using the RX_BUSYand XMIT_IDLE bits as shown in Table 35-11.

TABLE 35-11: PS/2 IDLE STATUS

RX_BUSY	XMIT_IDLE	Status
0	1	IDLE
1	X	NOT IDLE
X	0	

36.0 KEYBOARD MATRIX SCAN SUPPORT

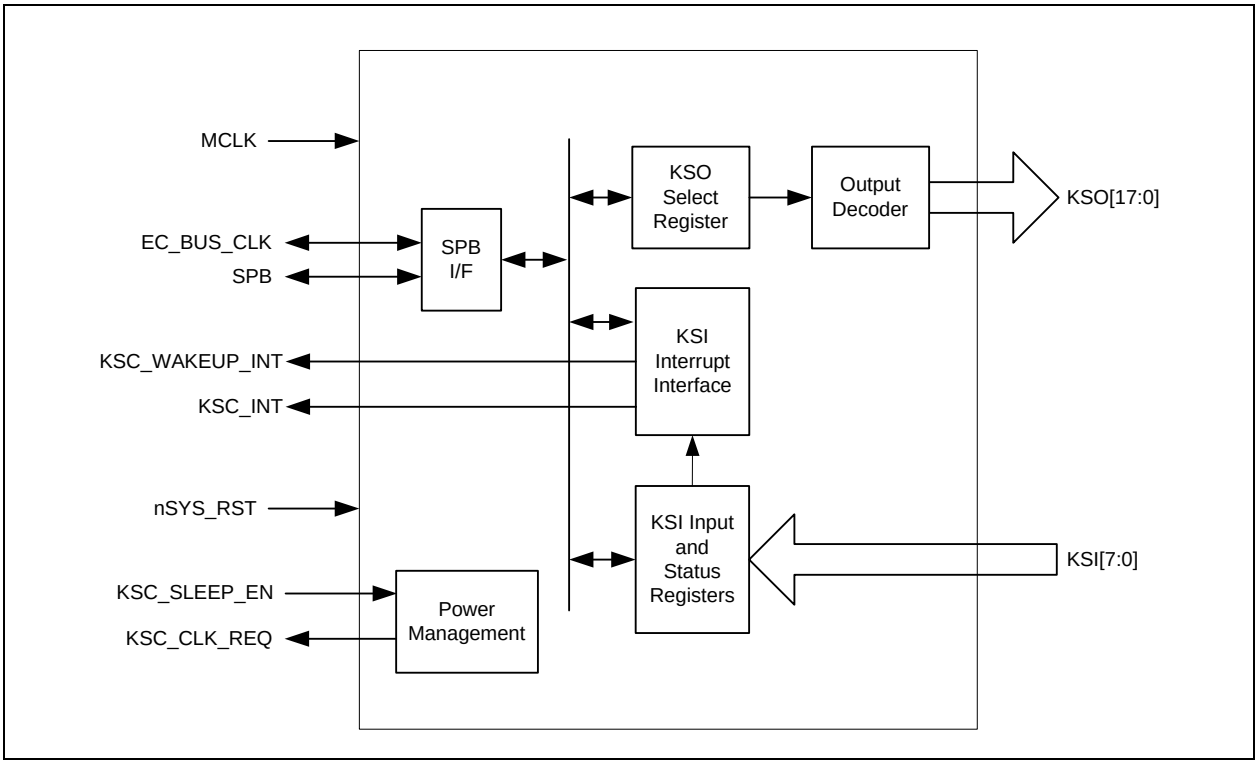
36.1 Overview

The Keyboard Scan Interface block provides a register interface to the EC to directly scan an external keyboard matrix of size up to 18x8. This block is attached to the EC SPB bus as EC Logical Device 8h.

APPLICATION NOTE: 18x8 is the maximum configuration. For smaller matrix size, firmware does not configure GPIO mux control of unused KSOs and masks out unused KSIs and associated interrupts.

36.2 Block Diagram

FIGURE 36-1: Keyboard Matrix Scan Support Block Diagram



36.3 Port List

TABLE 36-1: KEYBOARD SCAN INTERFACE SIGNAL LIST

Signal Name	Direction	Description
SPB	I/O Bus	EC MEC1609/MEC1609i peripheral bus
MCLK	INPUT	Master MEC1609/MEC1609i clock
EC Bus Clock	INPUT	SPB Bus Clock
nSYS_RST	INPUT	Block reset signal
KSI[7:0]	INPUT	Column inputs from external keyboard matrix. See Section 36.6
KSO[17:0]	OUTPUT	Row outputs to external keyboard matrix. See Section 36.6
KSC_INT	OUTPUT	Interrupt request to the Interrupt Aggregator's interrupt interface
KSC_WAKEUP_INT	OUTPUT	Wake-up request to the Interrupt Aggregator's wake-up interface
KSC_SLEEP_EN	INPUT	Sleep enable. See Section 36.5.
KSC_CLK_REQ	OUTPUT	Clock request. See Section 36.5.

36.4 Power, Clocks, and Resets

36.4.1 POWER DOMAIN

This block is powered by the VTR power supply.

36.4.2 CLOCKS

The block uses the EC Bus Clock and the 64.52-MHz MCLK.

36.4.3 RESET

The block is reset on assertion of nSYS_RST.

36.5 Power Management

The Keyboard Scan Interface KSC_CLK_REQ = ~KSC_SLEEP_EN.

36.6 Pins and I/O Buffers

26 pins are connected to the keyboard matrix. All are multiplexed with GPIOs and are configured using GPIO control registers (see Section 2.0, "Pin Configuration," on page 8 for details).

Row outputs: KSO[17:0] - 18 tri-state open-drain, 8 mA outputs; output well powered by VTR. When the block is disabled (KSEN = 1), KSO output buffers are tri-stated.

Column inputs: KSI[7:0] - 8 Schmitt trigger inputs with internal pull up; input well powered by VTR.

36.6.1 CONFIGURATION OF IO BUFFERS (INFORMATIONAL)

For KSO[17:0],

GPIO mux control bits are set to select KSO function.

Buffer type = open drain (KSOs from the Key Scan block can be used as output buffer enables; buffers are disabled if the block is disabled, i.e., when KSEN = 1).

Pull-up (if no external pull ups) or none (if external pull ups)

For KSI[7:0],

IO buffers can be configured either as signal function (KSI) or GPIO inputs. When the latter is selected, GPIO interrupts can be used in place of the block's interrupts.

36.7 Operation (Informational)

During scanning the firmware sequentially drives low one of the rows (KSO[17:0]) and then reads the column data line (KSI[7:0]). A key press is detected as a zero in the corresponding position in the matrix. Keys that are pressed are debounced by firmware. Once confirmed, the corresponding keycode is loaded into host data read buffer in the 8042 Host Interface module. Firmware may need to buffer keycodes in memory in case this interface is stalled or the host requests a Resend.

36.7.1 INTERRUPT GENERATION

To support interrupt-based processing, interrupt can optionally be generated on the high-to-low transition on any of the KSI inputs. Interrupts are to be registered without a running clock.

36.7.1.1 Runtime interrupt

KSC_INT output port 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 are 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. Interrupt is de-asserted when the status bit and/or interrupt enable bit is clear. A status bit cleared by writing '1' to it.

Interrupts from individual KSIs are logically ORed together to drive the KSC_INT output port. Once asserted, interrupt is not asserted again until either all KSI[7:0] have returned high or the KSO Driver Select[4:0] has changed.

36.7.1.2 Wake-up interrupt

KSC_WAKEUP_INT is the block's wakeup interrupt. It is routed to of the Interrupt Aggregator, KEYSCAN bit of the GIRQ18 Source Register.

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_WAKEUP_INT to be asserted. Also set is the associated status bit in the EC Blocks Clock Required Status Register 2. KSC_WAKEUP_INT remains active until the bus clock is started.

The aforementioned transition on KSI also sets 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.

36.8 Registers

36.8.1 REGISTERS SUMMARY

The Keyboard Scan Interface is assigned EC Logical Device Number 8h, which has AHB base address F0_2000h.

TABLE 36-2: KSC REGISTER SUMMARY

Base Address: F0_2000h	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Reserved	0h	3-0	R	
KSO Select Register	4h	0	R/W	
KSI Input Register	8h	0	R	
KSI Status Register	Ch	0	R/WC	
KSI Interrupt Enable Register	10h	0	R/W	

36.8.2 KSO SELECT REGISTER

TABLE 36-3: KSO SELECT REGISTER

HOST OFFSET	**N/A**			N/A			**HOST SIZE**	
EC OFFSET	**4h (R/W)**				32-bit		**EC SIZE**	
POWER	VTR			0...040h			**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	KSO INVERT	KSEN	KSO ALL	KSO Driver Select[4:0]				

KSO INVERT

KSO INVERT = 1 inverts KSO[17:0]. When KSO INVERT = 0 KSO[17:0] operate normally See Table 36-5, "Keyboard Scan Out Control Summary," on page 462.

KSEN

KSEN = 1 disables keyboard drives. See first row in Table 36-5. KSEN = 0 enables keyboard scan

KSO ALL

KSO ALL = 1, drives all KSO lines according to KSO INVERT bit. See Table 36-5, “Keyboard Scan Out Control Summary,” on page 462.

KSO DRIVER SELECT[4:0]

KSO Driver Select[4:0] controls the corresponding KSO line (00000b = KSO[0] etc.) according to KSO INVERT. See Table 36-4, "KSO Select Decode".

TABLE 36-4: 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
0Bh	KSO11
0Ch	KSO12
0Dh	KSO13
0Eh	KSO14
0Fh	KSO15
10h	KSO16
11h	KSO17

TABLE 36-5: KEYBOARD SCAN OUT CONTROL SUMMARY

D7 KSO Invert	D6 KSEN	D5 KSO ALL	D[4:0] KSO Drivers Address	Description
X	1	x	x	Keyboard Scan disabled. KSO[17:0] output buffers disabled.
0	0	0	10001b-00000b	KSO[Drive Selected] driven low. All others driven high
1	0	0	10001b-00000b	KSO[Drive Selected] driven high. All others driven low
0	0	0	11111b-10010b	ALL KSO’s driven high
1	0	0	11111b-10010b	All KSO’s driven low
0	0	1	x	KSO[17:0] driven low
1	0	1	x	KSO[17:0] driven high

36.8.3 KSI INPUT REGISTER

TABLE 36-6: KSI INPUT REGISTER

BUS OFFSET	**N/A**				N/A		**HOST SIZE**	
EC OFFSET	**8h (R)**				32-bit		**EC SIZE**	
POWER	VTR				0...00h		**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	KS7	KS6	KS5	KS4	KS3	KS2	KS1	KS0

36.8.4 KSI STATUS REGISTER

TABLE 36-7: KSI STATUS REGISTER

HOST BUS OFFSET	**N/A**				N/A		**HOST SIZE**	
EC OFFSET	**Ch (R/WC)**				32-bit		**HOST SIZE**	
POWER	VTR				0...00h		**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	KSI[7] STAT	KSI[6] STAT	KSI[5] STAT	KSI[4] STAT	KSI[3] STAT	KSI[2] STAT	KSI[1] STAT	KSI[0] STAT

BIT DEFINITION:

KSI[x] STAT bit is set on the falling edge of the corresponding KSI input. Writing a 1 to a bit will clear it.

A KSI interrupt is generated when its corresponding status bit and interrupt enable bit are both set. KSI interrupts are logically ORed together to produce KSC_INT and KSC_WAKEUP_INT.

36.8.5 KSI INTERRUPT ENABLE REGISTER

TABLE 36-8: KSI INTERRUPT ENABLE REGISTER

HOST BUS OFFSET	**N/A**				N/A			**HOST SIZE**
EC OFFSET	**10h (R/W)**				32-bit			**HOST SIZE**
POWER	VTR				0...00h			**VTR POR DEFAULT**
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	KSI[7] IntEn	KSI[6] IntEn	KSI[5] IntEn	KSI[4] IntEn	KSI[3] IntEn	KSI[2] IntEn	KSI[1] IntEn	KSI[0] IntEn

BIT DEFINITION:

KSI[x] IntEn enables interrupt generation due to high-to-low transition on KSI[x] input. An interrupt is generated only when both KSI[x] IntEn and KSI[x] STAT bits are set.

37.0 BC-LINK MASTER

37.1 General Description

The function of this block is to.provide BC-Link to a slave device. The BC-Link protocol includes a start bit to signal the beginning of a message and a turnaround (TAR) period for bus transfer between the Master and Companion devices.

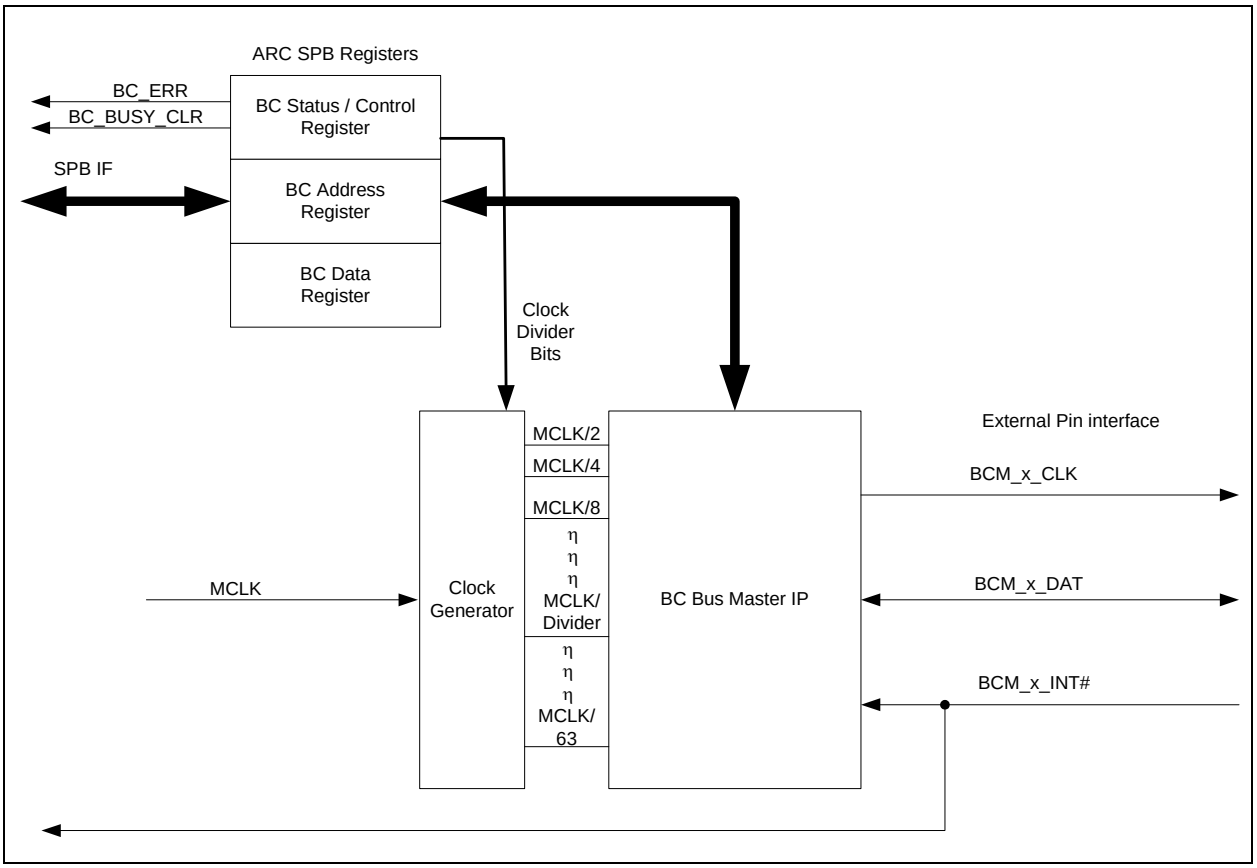
Note:	A weak pull-up resistor is recommended on the data line (100KΩ).

There are four instances of the BC-Link Master interface in the MEC1609/MEC1609i. Instance A, B, and C are high speed BC-links with high speed buffers attached. Instance D is a low speed BC-Link which uses 8 mA buffer. Instance D consists of signals LSBCM_D_INT#, LSBCM_D_DAT, LSBCM_D_CLK.

The maximum usable clock frequencies are described in Note 37-1 on page 466.

37.2 Block Diagram

FIGURE 37-1: BC-LINK MASTER BLOCK DIAGRAM



37.3 Signal List

TABLE 37-1: BC-LINK SIGNAL LIST

Signal Name	Direction	Description
BCM_x_CLK	OUTPUT	64MHz - 252KHz output clock, where x is A, B,C or D.
BCM_x_DAT	INPUT/OUTPUT	Bidirectional data line, where x is A, B,C or D.
BCM_x_INT#	INPUT	Input from the companion device, where x is A, B,C or D.

TABLE 37-1: BC-LINK SIGNAL LIST (CONTINUED)

Signal Name	Direction	Description
BC_x_ERR	OUTPUT	BC-Link master error interrupt
BC_x_BUSY_CLR	OUTPUT	BC-Link master Busy Clear interrupt
MCLK	INPUT	64 MHz Master clock
SPB	I/O Bus	Bus used for register access
CLK_REQ	OUTPUT	Power Management clock required output.

Note: The Low Speed BC-Link Master Block has an identical organization, but the external signals are LSBCM_D_CLK, LSBCM_D_DAT and LSBCM_D_INT#.

See Table 2-8, “BC-Link Interface,” on page 14 for the pin interface and associated Note 2-1 & Note 2-2 on page 14.

37.4 Power, Clocks and Reset

37.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.5, "Power Configuration," on page 93 for details on power domains.

37.4.2 CLOCKS

This block uses the EC Bus Clock and the 64.52MHz (MCLK). The EC Bus Clock is used to access the Registers described in this block. MCLK is divided down to generate the external bus clock.

See Section 5.4, "Clock Generator," on page 76 for details on clocks.

37.4.3 POWER ON RESET

This block is reset on a nSYS_RST. On reset, the BC-Link state machine transitions to the Idle state and waits for the address and data registers to be written.

See Section 5.6, "Reset Interface," on page 95 for details on reset.

37.5 Interrupts

Each BC-Link Master instance has three interrupts events: BC_BUSY_CLR, BC_ERR, & BC_INT#. The BC-Link Master BC_BUSY_CLR and BC_ERR interrupts are generated by changes in the BC-Link Status Register. The BC_INT# is an active low level interrupt generated by input pin signal function. The edge detection of the interrupt and wake events are controlled by their associated pin control registers in the Section 22.0, "GPIO Interface," on page 329

The BC-Link Master block Instance A are routed to the BCM_BUSY_CLR[A], BCM_ERR[A], & BCM_INT#[A] bits in the GIRQ21 Source Register on page 284. The BC-Link Master block Instance B are routed to the BCM_BUSY_CLR[B], BCM_ERR[B], & BCM_INT#[B] bits in the GIRQ21 Source Register on page 284. The BC-Link Master block Instance C are routed to the BCM_BUSY_CLR[C], BCM_ERR[C], & BCM_INT#[C] bits in the GIRQ21 Source Register on page 284. The BC-Link Master block Instance D are routed to the BCM_BUSY_CLR[D], BCM_ERR[D], & BCM_INT#[D] bits in the GIRQ21 Source Register on page 284.

37.5.1 INSTANCE BUFFERS VS CLOCK

There are four instances of the BC-Link Master block implemented in the MEC1609/MEC1609i enumerated as A, B, C, D. Table 2-8, “BC-Link Interface,” on page 14 lists the pin functions for all instances. The base addresses for all instances are listed in Table 37-3, "BC-Link Master Base Address Table".

Instance A, B, & C are high speed BC-links with higher current buffers driving the external signals. Instance D is a low speed BC-Link, which is implemented lower current buffers. Signal names for the high speed interfaces are BCM[A,B,C]_CLK, BCM[A,B,C]_DAT and BCM[A,B,C]]_INT#. Signal names for the low speed instance are LSBCM_D_CLK, LSBCM_D_DAT and LSBCM_D_INT#.

Note 37-1 For ribbon cable applications, the Low Speed BC-Link Master maximum clock frequency is 3Mhz. The High Speed BC-Link Master maximum clock frequency is 21.41 Mhz. The Clock frequency is set with the BC-Link Clock Select Register. See Table 37-2 on page 467. and Note 41-3 on page 516.

TABLE 37-2: BC-Link Master PIN INTERFACE

Pin	Description	Buffer	Max Freq	Min Value in BC-Link Clock Select Register
BCM[A,B,C]_CLK	High Speed Clock supplied by the MEC1609/MEC1609i Device	O16	21.41Mhz	2
BCM[A,B,C]_DAT	High Speed Data Line	IO16 Note 37-2		
BCM[A,B,C]_INT#	Interrupt signal	I		
LSBCM_D_CLK	Low Speed Clock supplied by the MEC1609/MEC1609i Device	O16	3MHz	21
LSBCM_D_DAT	Data Line	IO8 Note 37-2		
LSBCM_D_INT	Interrupt signal	I		

Note 37-2 BCM[A,B,C]_DAT & LSBCM_D_DAT pins requires a weak pull up (100K).

37.6 Registers

TABLE 37-3: BC-Link Master BASE ADDRESS TABLE

BC-Link InstanceS	LDN from (Table 3-2 on page 48)	AHB Base Address
BC-LINK.A	5h	F0_1400h
BC-LINK.B		F0_1480h = F0_1400h + 80h
BC-Link.C		F0_1500h = F0_1400h + 100h
BC-Link.D		F0_1580h = F0_1400h + 180h

TABLE 37-4: BC-Link Master REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
BC-Link Status Register	00h	0	R/W	
BC-Link Address Register	04h	0	R/W	
BC-Link Data Register	08h	0	R/W	
BC-Link Clock Select Register	0Ch	0	R/W	

37.6.1 BC-LINK STATUS

TABLE 37-5: BC-LINK STATUS REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	00h						8-bit	**EC SIZE**
POWER	VTR						81h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-‘	-	-	-	-
EC TYPE	R/W	R/WC	R/W	R/W	R	R	R	R
BIT NAME	RESET	BC_ ERR	BC_ ERR_ INT_EN	BC_ Busy_ CLR_ INT_EN	Reserved			BUSY

BUSY

This bit is asserted to ‘1’ when the BC interface is transferring data and on reset. Otherwise it is cleared to ‘0’. When BUSY bit is cleared by hardware, an interrupt is generated if the BC_Busy_CLR_INT_EN bit is set to ‘1’.

BC_BUSY_CLR_INT_EN

This bit is an enable for generating an interrupt when the BUSY bit is cleared by hardware. When the BC_Busy_CLR_INT_EN bit is set to ‘1’, the interrupt signal is enabled. When the BC_Busy_CLR_INT_EN bit is cleared to ‘0’, the interrupt is disabled. When enabled this interrupt occurs after a BC Bus read or write. Figure 37-2 shows when the interrupt is generated.

BC_ERR_INT_EN

This bit is an enable for generating an interrupt when the BC_ERR bit is set by hardware. When the BC_ERR_INT_EN bit is ‘1’, the interrupt signal is enabled. When the BC_ERR_INT_EN bit is ‘0’,the interrupt is disabled.

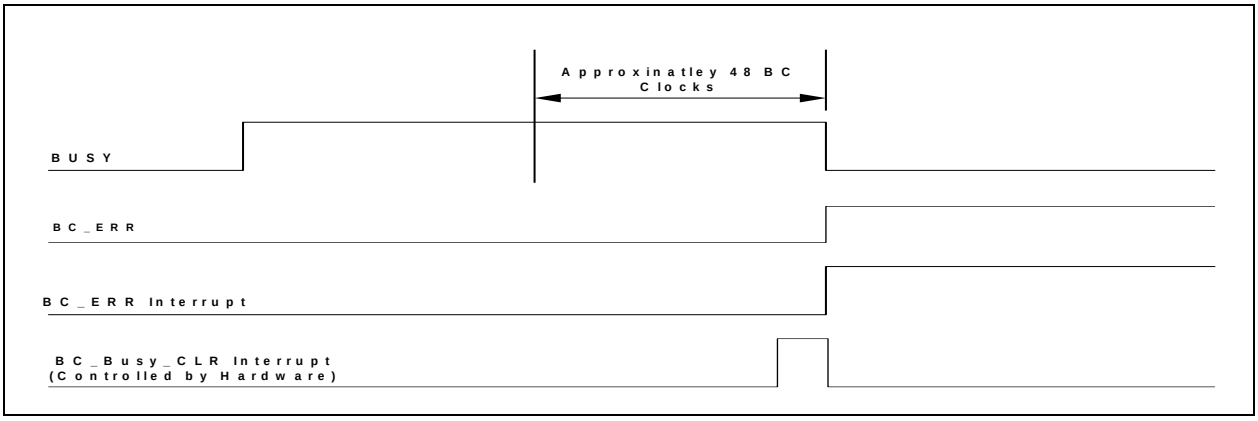
BC_ERR

This bit indicates that a BC Bus Error has occurred. If an error occurs the BC_ERR bit is set by hardware when the BUSY bit is cleared. See Figure 37-2 for relative timing. When BC_ERR bit is set by hardware, an interrupt is generated if the BC_ERR_INT_EN bit is set to ‘1’.

This BC_ERR bit is cleared to ‘0’, by software writing a ‘1’ to this bit.

Errors that cause this interrupt are: Bad Data received by the BASE (CRC Error) or a time-out caused by the COMPANION not responding. All COMPANION errors cause the COMPANION to abort the operation and cause the BASE to time-out. Figure 37-2 shows the timing of this interrupt.

FIGURE 37-2: BC BUS BC_ERR INTERRUPT TIMING



RESET

When set to ‘1’,the BC-Link Master Interface will be placed in reset and be held in reset until this bit is de-asserted. Reset causes the BUSY bit to be set and will not be cleared until the reset operation of the BC Interface is completed (approximately 48 BC clocks).

PROGRAMMER’S NOTE: The de-assertion of the BUSY bit on reset will not generate an interrupt, even if the BC_Busy_CLR_INT_EN is asserted 1. The BUSY bit must be polled.

37.6.2 BC-LINK ADDRESS

TABLE 37-6: BC-LINK ADDRESS REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	04h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Address[7:0]							

37.6.3 BC-LINK DATA

TABLE 37-7: BC-LINK DATA REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	08h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data[7:0]							

37.6.4 BC CLOCK SELECT

TABLE 37-8: BC-LINK CLOCK SELECT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	0Ch						8-bit	**EC SIZE**
POWER	VTR						4	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Divider[7:0]							

DIVIDER

The BC Clock is set to the Master Clock divided by this field, or MCLK/ (Divider[7:0] +1), Where Divider[7:0] is 0 to 255. The clock divider bits can only can be changed when the BC Bus is in soft RESET (when either the Reset bit is set by software or when the BUSY Bit is set by the interface).

Example settings are shown in Table 37-9, "Example Frequency Settings":

TABLE 37-9: EXAMPLE FREQUENCY SETTINGS

Divider	Frequency
0	64.52MHz
1	32.25MHz
2	21.41MHz
3	16.13MHz
4	12.9MHz
15	4.033MHz
21	2.99MHz
2A	1.50 MHz
63	1MHz

37.7 BC-Link Master Operations

Descriptions of the BC-Link read and write operations follows:

37.7.1 READ

The BC-Link Read protocol requires two reads of the BC-Link Data Register. The two reads drive a two state-state machine: the two states are Read#1 and Read#2. The Read#1 of the BC-Link Data Register starts the read protocol on the BC-Link pins and sets the Busy bit in the BC-Link Status Register. The contents of the data read during Read#1 by the EC is stale and is not to be used. After the Busy bit in the BC-Link Status Register autonomously clears to '0', the Read#2 of the BC-Link Data Register transfers the data read from the peripheral/BC-Link companion chip to the EC.

1. Software starts by checking the status of the BUSY bit in the BC-Link Status Register on page 468. If the Busy bit is 0, proceed, if Busy is a 1 Wait.
2. Software writes the address of the register to be read into the BC-Link Address Register on page 469.
3. Software then reads the BC-Link Data Register on page 470. This read returns random data. The read activates the BC-Link Master to transmit the read request packet to the BC-Link companion. When the transfer initiates, the hardware sets the BUSY bit to a 1.
4. The BC-Link companion reads the selected register and transmits the read response packet to theBC-Link Master.

   Note 37-1 The companion will ignore the read request if there is a CRC error, this will cause the base to time-out and issue a BC_ERR Interrupt.

5. The BC-Link Master loads the BC-Link Data Register on page 470 issues a BUSY Bit Clear interrupt and clears the busy bit to '0'.
6. Check the BC_ERR bit in the BC-Link Status Register on page 468.
7. Software can now read the BC-Link Data Register on page 470 which contains the valid data if there was no BC Bus error.
8. If a Bus Error occurs issue a soft reset by setting the Reset bit in the BC-Link Status Register on page 468.
9. The read can re-tried once BUSY is cleared.

PROGRAMMER'S NOTE: Steps 3 thorough 7 should be completed as a contiguous sequence. If not the LSBC interface could be presenting incorrect data when software thinks it is accessing a valid register read.

37.7.2 WRITE

1. Software starts by checking the status of the BUSY bit in the BC-Link Status Register on page 468. If the BUSY bit is 0, proceed, if BUSY is a 1 Wait.
2. Software writes the address of the register to be written into the BC-Link Address Register on page 469. Then writes the data to be written into the addressed register in to the BC-Link Data Register on page 470.
3. The write to the BC-Link Data Register on page 470 starts the BC_Link write operation. The BC-Link Master sets the BUSY bit in the BC-Link Status Register on page 468.
4. The BC-Link Master Interface transmits the write request packet.
5. When the write request packet is received by the BC-Link companion, the CRC is checked and data is written to the addressed companion register.
6. The companion sends an ACK if the write is completed.

   Note 37-2 A time-out will occur approximately 16 BC-Link clocks after the packet is sent by the BC-Link Master. The BC-Link Master will issue a BC_ERR bit in the BC-Link Status Register on page 468 approximately 48 clocks later. and clear the BUSY bit.

7. The BC-Link Master issues the Busy bit Clear interrupt and clears the BUSY bit after receiving the ACK from the companion
8. If a Bus Error occurs issue a soft reset by setting the Reset bit in the BC-Link Status Register on page 468.
9. The write can re-tried once BUSY is cleared.

37.8 Power Management

BC-Link Master Power Management is illustrated in Table 37-10. Note that the BC-Link Master does not include a sleep enable input.

TABLE 37-10: BC-Link Master Power Management

Block Enable (Note 37-3)	Block Idle Status (Note 37-4)	Clock Required Status Output (CLK_REQ)	State	Description
DISABLED	X	0	DISABLED	Block is disabled by firmware and the core clock is not needed.
ENABLED	NOT IDLE	1	NORMAL OPERATION	The block is enabled and performing a transaction.
	IDLE	0	SLEEPING	The block is enabled and not performing a transaction. The clock may be stopped.

Note 37-3 The DISABLED state in Table 37-10 is defined as the Reset Bit being set to '1' and the & BUSY bit being cleared to'0' in the BC-Link Status Register.

Note 37-4 The IDLE state in Table 37-10 is defined as both the Reset & BUSY Bit being cleared to '0' in the BC-Link Status Register.

38.0 SERIAL DEBUG PORT

38.1 General Description

The Serial Debug Port serially transmits MCU-originated diagnostic vectors to an external debug trace system.

The Serial Debug Port consists of the Debug Data Register, Debug Control Register, a Parallel-to-Serial Converter, a Clock/Control Interface and a two-pin external interface (Debug_CLK, Debug_DAT). See Figure 38-1.

38.2 Power, Clocks and Reset

38.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 5.1.1, "Power Configuration," on page 73 for details on power domains.

38.2.2 CLOCKS

This block has one clock input, the EC Bus Clock.

See Section 5.1.2, "Clock Generator," on page 73 for details on clocks.

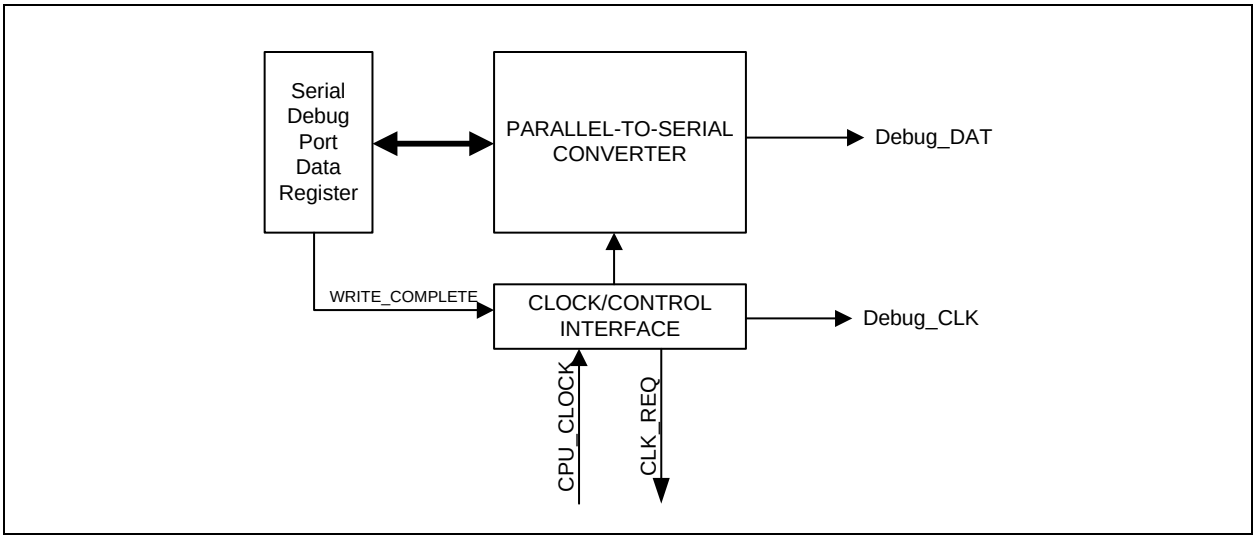
38.2.3 RESET

This block is reset on a nSYS_RST.

See Section 5.1.3, "Reset Interface," on page 73 for details on reset.

38.3 Block Diagram

FIGURE 38-1: SERIAL DEBUG PORT BLOCK DIAGRAM



38.4 Block Diagram Port List

TABLE 38-1: Serial Debug Port PORT LIST

Signal Name	Direction	Description
Debug Clock	OUTPUT	Derived from the EC Bus Clock
Debug Data	OUTPUT	Serialized Data shifter out by the Debug Clock
EC Bus Clock	INPUT	EC AHB Bus Clock
CLK_REQ	OUTPUT	Power Management Clock Required output signal

38.5 Interrupts

There are no interrupts from this block.

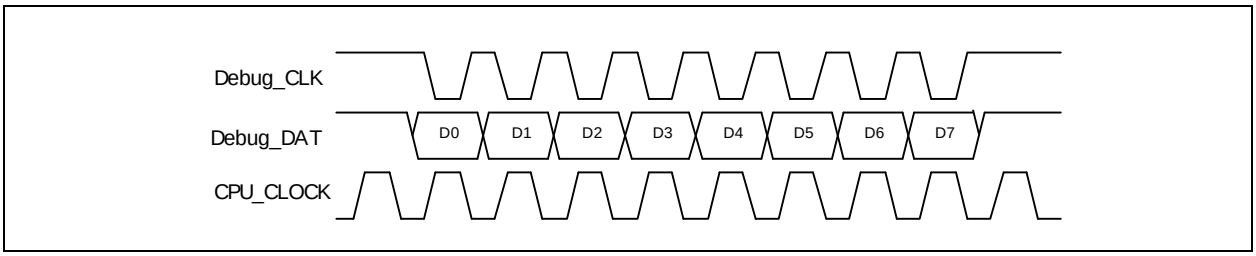
38.6 Functional Description

Writes to the Debug Data Register initiate an transfer cycle (Figure 38-2). Data from the Debug Data Register is shifted LSB first and is transmitted one byte per transfer cycle.

Data is transferred in one direction only from the Debug Data Register to the external interface. The data is shifted out on the clock edge selected by the EDGE_SEL bit in the Debug Control Register on page 475. Valid data is provided on opposite edge of Debug_CLK after being shifted out. For example when the EDGE_SEL bit is '0' (default), Valid data is provided on the falling edge of Debug_CLK. The Setup Time to the falling edge of Debug_CLK is 10ns minimum; the Hold Time is 1ns minimum.

The Debug_CLK and Debug_DAT outputs are '1' when the serial Debug Port is inactive. The transfer clock is EC Bus Clock. See also Section 41.12, "Serial Debug Port Timing," on page 524.

FIGURE 38-2: DATA TRANSFER



38.7 Instance Description

There is one Serial Debug Port instance defined in this chapter.

38.8 Registers

Each instance of the Serial Debug Port has its own Logical Device Number, and Base Address as indicated in Table 38-2.

TABLE 38-2: Serial Debug Port BASE ADDRESS TABLE

Serial Debug Port Instance	LDN from (Table 3-4 on page 49)	AHB Base Address
MCU Debug Port	23h	F0_8C00h

The following table summarizes the registers allocated for the Serial Debug Port. The offset field in the following table is the offset from the Serial Debug Port's EC Base Address.

TABLE 38-3: Serial Debug Port REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Debug Data Register	00h	0	R/W	
Debug Control Register	04h	0	R/W	

38.8.1 DETAILED REGISTER DESCRIPTIONS

The Debug Data Register is R/W. It always returns the last data written by the MCU or the power-on default '00h'.

TABLE 38-4: DEBUG DATA REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	00h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	--	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data[7:0]							

DATA[7:0]

Debug data to be shifted out on the MCU Debug Port. While data is being shifted out, the SPB interface will 'hold-off' additional writes to the data register until the transfer is complete.

TABLE 38-5: DEBUG CONTROL REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	04h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	--	-	-	-	-
EC TYPE	R						R/W	R/W
BIT NAME	Reserved						EN	EDGE_SEL

EDGE_SEL

0= Data is shifted out on the rising edge of the debug clock (Default)

1= Data is shifted out on the falling edge of the debug clock

EN

0= Clock is disabled (Default)

1= Clock enabled

Note: The EN bit must not be de-asserted in the middle of a transfer.

38.9 Power Management

Serial Debug Port Power Management is illustrated in Table 38-6. Note that the Serial Debug Port does not include a sleep enable input.

TABLE 38-6: Serial Debug Port Power Management

Block Enable (Note 38-1)	Block Idle Status (Note 38-2)	Clock Required Status (CLK_REQ)	State	Description
DISABLED	X	0	DISABLED	Serial Debug Port is disabled by firmware and the clock is not needed.
ENABLED	NOT IDLE	1	NORMAL OPERATION	The Serial Debug Port is enabled and performing a transaction.
	IDLE	0	SLEEPING	The Serial Debug Port is enabled and not performing a transaction. The clock may be stopped.

Note 38-1 The Block Enable in Table 38-6 is defined as the EN Bit in the Debug Control Register.

Note 38-2 The Serial Debug Port is ‘idle’ when not involved in a data transfer. There is no registered bit in the Serial Debug Port interface to indicate the idle state.

39.0 JTAG AND XNOR

39.1 General Description

The MEC1609/MEC1609i includes a JTAG Slave for Debugging ARC Firmware, a Boundary Scan slave, and a JTAG Master. All of these function share the MEC1609/MEC1609i JTAG Interface as defined in Section 2.4.4.

The JTAG slaves are asynchronously reset and deactivated when the JTAG_RST# input pin is asserted ('0'). When JTAG_RST# is not asserted, only one slave is enabled. JTAG Slave Selection depends on the state of the TAP Controller Select Strap Option. The JTAG Master is independent of the slave TAP controllers and cannot be used when the JTAG_RST# pin is not asserted.

39.2 Slave Selection

39.2.1 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 (see Section 2.7, "Strapping Option," on page 41) is sampled by hardware at VTR POR according to the Slave Select Timing as defined in Section 39.2.2 and is registered internally to select between the debug and boundary scan TAP controllers.

If the TAP Controller Select Strap Option 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.

39.2.2 SLAVE SELECT TIMING

The JTAG_RST# input pin must be asserted at VTR power-up and follow the timing as defined in this section. The relationship between the TAP Controller Select Strap Option sample timing and the JTAG_RST# pin de-assertion timing is illustrated below in Figure 39-1 and Table 39-1.

FIGURE 39-1: ASYNCHRONOUS JTAG RESET AND TAP Controller Select Strap Option POWER-UP TIMING

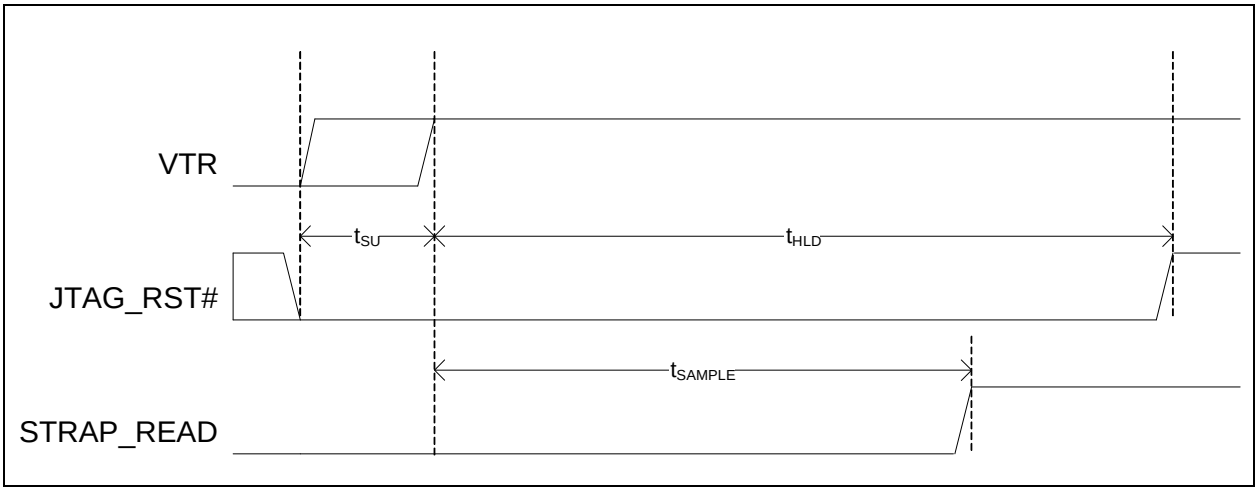


TABLE 39-1: Slave Select Timing PARAMETERS

Parameters	Symbol	MIN	Units	Notes
JTAG_RST# Setup Time	t_{SU}	0	sec	This is a requirement from the IEEE 1149.1 spec.
JTAG_RST# Hold Time	t_{HLD}	ARC reset time + 1 us	–	t_{SAMPLE} is the same as $t_{STRETCH}$ in FIGURE 5-10: VTR Power-Up Timing on page 97.
TAP Controller Select Strap Option Sample Time	t_{SAMPLE}	ARC reset time	–	

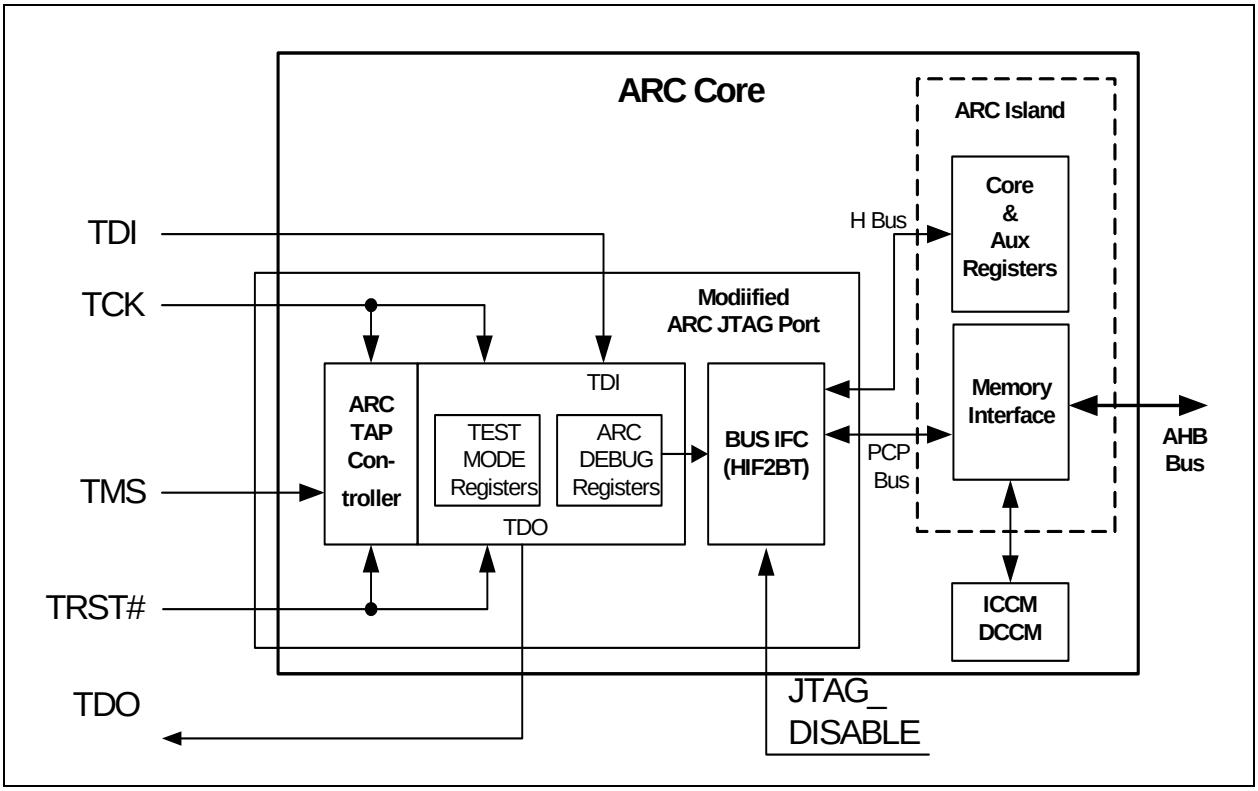
39.3 JTAG Slave for Debugging ARC Firmware

The ARC JTAG Port is defined in the ARC 600 External Interfaces Reference Manual, Chapter 2. The Microchip modifications are described in this chapter. The ARC JTAG Port has been modified by Microchip to provide additional Data Registers (see Note 39-1.) The TEST MODE Register provide additional on-chip support specific to the MEC1609/MEC1609i.

39.3.1 ARC JTAG CAPABILITIES

- Fully compliant with IEEE1149.1 standard
- 4-bit Instruction Register
- Standard 1-bit BYPASS register
- Standard 32-bit IDCODE register
- Four JTAG registers give access to on-chip memory and register resources
- Can read or write a 32-bit quantity from or to any ARC Core Register, Aux Register or 32-bit aligned memory location. No other interfaces are provided or needed
- Accesses to Aux Registers and Memory do not require the ARC processor to be halted
- Memory accesses are always performed in units of 32 bits

FIGURE 39-2: BLOCK DIAGRAM OF ARC JTAG SYSTEM



As shown in Figure 39-2, the ARC JTAG Port sits within the ARC Core, between the JTAG signals and the ARC Island block, which contains the Core and Aux register sets and the Memory Interface to the Closely-Coupled Memories (ICCM / DCCM) and the AHB Bus (FLASH memory and Memory-Mapped I/O registers). It is not associated with any form of Boundary Scan, but instead is used by an external JTAG host to access memory and register resources on behalf of a debugger program. There are no other connections between the ARC JTAG Port and the rest of the ARC Core: register access is enough to halt and restart the processor, as well as to detect that the processor has halted (a continuous poll on the STATUS32 Aux Register).

Internally, the ARC JTAG port consists of the following sub-blocks:

- The ARC Test Access Port (TAP) Controller. This block is driven by the JTAG_CLK, TMS and JTAG_RST# inputs, and provides the control signals for the JTAG data transfers. It is a single state machine consisting of 16 states, whose transitions are controlled strictly by the state of TMS on each rising edge of JTAG_CLK. The low-active JTAG_RST# input provides an asynchronous reset, though JTAG_CLK and TMS together can also bring the TAP Controller to the reset state (5 consecutive JTAG_CLK rising edges with TMS=1).
- The ARC DEBUG & TEST MODE Registers blocks. This blocks handles the data transfers as directed by the TAP, and contains the registers and shift registers internal to the JTAG Port. The holding registers in this block consist of a 4-bit Instruction register, a 1-bit Bypass register, and a set of Data registers (see Note 39-1) of various lengths. There are three sets of Data Registers: JTAG Standard Data Registers, JTAG Debug Data Registers, & JTAG Test Mode Data Registers. The JTAG Test Mode Data Registers provide additional on-chip support specific to the MEC1609/MEC1609i.
- The Bus Interface block. This block accepts values for the JTAG Debug Data Registers: ADDRESS, DATA, STATUS and TRANSACTION COMMAND, and uses them to request data transfers from the ARC's inner core ("Island") sub-block.
- As part of the Boot block protection the JTAG Port's JTAG Debug Data Registers can be disabled to prevent an external debugger from potentially halting the EC and then reading or re programing any word in the Boot Block.

39.4 Boundary Scan

Boundary Scan includes registers and functionality as defined in IEEE 1149.1 and the MEC1609/MEC1609i BSDL file. Functionality implemented beyond the standard definition is summarized in Table 39-2. The MEC1609/MEC1609i JTAG ID is 02002445h.

TABLE 39-2: EXTENDED Boundary Scan FUNCTIONALITY

	Control Bit	Function	Description
1.	userDRbit0	Double Bond Testing	When userDRbit0 is high, tap override is enabled.
2.	userDRbit1		'1' = Selects PECI. '0' = Selects GPIOs.
3.	userDRbit2	TAP Controller Select Strap Option Override	When userDRbit2 is '1,' TAP Controller Select Strap Option is overridden to select the debug TAP Controller until the next time that the TAP Controller Select Strap Option is sampled.

39.5 JTAG Master

39.5.1 OVERVIEW

The JTAG Master controller in the MEC1609/MEC1609i 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.

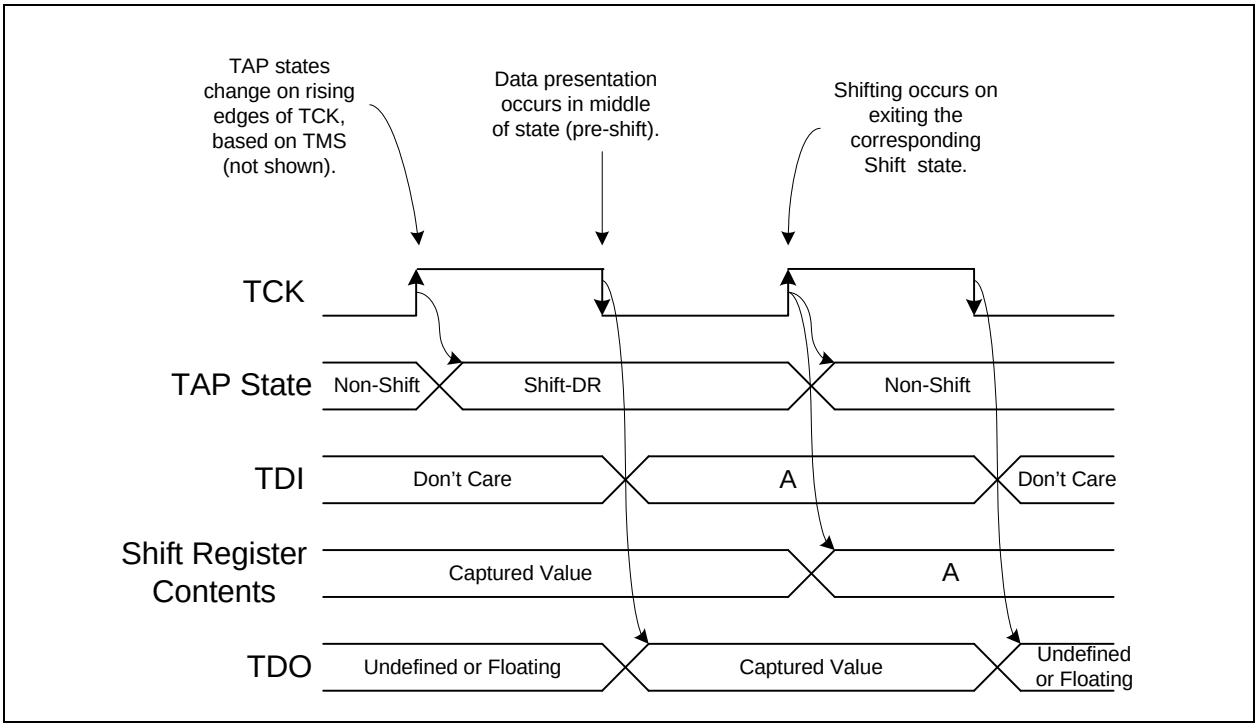
The JTAG Master interface shares the JTAG pin interface with the Boundary Scan and Debug TAP controllers; including, JTAG_CLK, JTAG_TDI, JTAG_TDO and JTAG_TMS. When the MEC1609 JTAG interface is configured as master, it is the responsibility of the master firmware to satisfy all requirements regarding 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.

39.5.2 DESCRIPTION

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 39-3, "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.

FIGURE 39-3: JTAG SIGNAL CLOCKING



39.5.3 JTAG MASTER REGISTER INTERFACE

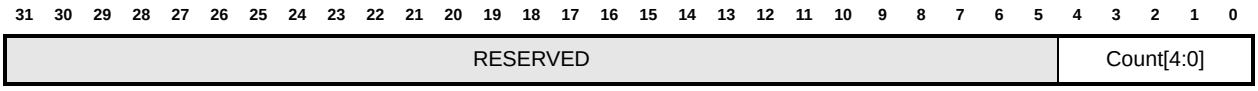
The JTAG Master interface uses the ARC Auxiliary Register interface; all JTAG Maser registers are Auxiliary registers, accessed by the ARC `lr` (load auxiliary register) and `sr` (store auxiliary register) instructions. Table 39-3, "JTAG Auxiliary Registers" lists these registers:

TABLE 39-3: JTAG AUXILIARY REGISTERS

Register Number	Auxiliary Name	LR/SR R/W	VTR Default	Description
FFFF_FFFFh	JTAG_COM	W	0000_000h	JTAG Master Command Register
FFFF_FFFEh	JTAG_TMS	R/W	0000_000h	JTAG Master register source for TMS pin
FFFF_FFFDh	JTAG_TDI	R/W	0000_000h	JTAG Master register source for TDI pin
FFFF_FFFCh	JTAG_TDO	R/W	0000_000h	JTAG Master register destination for TDO pin
FFFF_FFFBh	JTAG_STATUS	R	0000_000h	JTAG Master Status register
FFFF_FFFAh	JTAG_CONFIG	R/W	0000_000h	JTAG Master Configuration register

39.5.3.1 JTAG Master Command, JTAG_COM

FIGURE 39-4: JTAG_COM



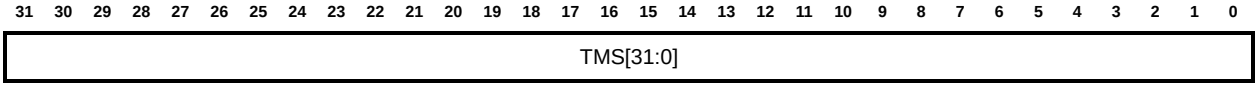
COUNT

If M in JTAG_CONFIG is 1, writing this field starts, clocking and shifting on the JTAG port. The JTAG Master port will shift count+1 times, so writing a 0 will shift 1 bit and writing 31 will shift 32 bits. The signal JTAG_CLK will cycle count+1 times. JTAG_TMS and JTAG_TDI will be shifted out on the falling edge of JTAG_CLK and JTAG_TDO will get shifted in on the rising edge of JTAG_CLK.

When M in JTAG_CONFIG is 0 the JTAG port is configured as a Slave and writing this field has no effect.

39.5.3.2 JTAG Master TMS, JTAG_TMS

FIGURE 39-5: JTAG_TMS

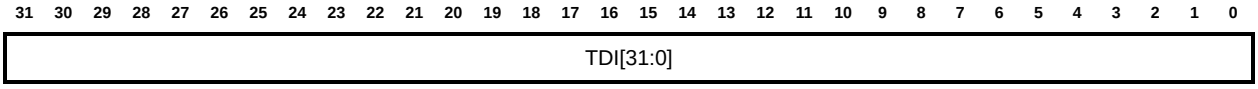


TMS

When JTAG_COM is written, from 1 to 32 bits are shifted out of TMS, starting with bit 0, onto the JTAG_TMS pin. Shifting is at the rate determined by CLK in JTAG_CONFIG.

39.5.3.3 JTAG Master TDI, JTAG_TDI

FIGURE 39-6: JTAG_TDI

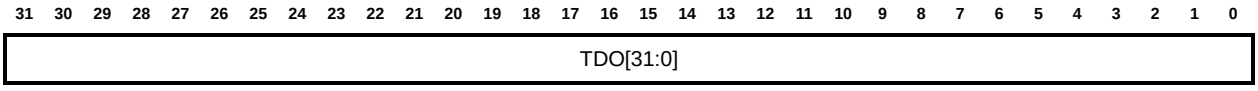


TDI

When JTAG_COM is written, from 1 to 32 bits are shifted out of TDI, starting with bit 0, onto the JTAG_TDI pin. Shifting is at the rate determined by CLK in JTAG_CONFIG.

39.5.3.4 JTAG Master TDO, JTAG_TDO

FIGURE 39-7: JTAG_TDO

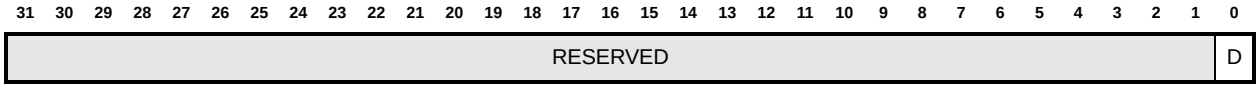


TDO

When JTAG_COM is written, from 1 to 32 bits are shifted into of TDO, starting with bit 0, onto the JTAG_TDO pin. Shifting is at the rate determined by CLK in JTAG_CONFIG.

39.5.3.5 JTAG Master Status, JTAG_STATUS

FIGURE 39-8: JTAG_STATUS



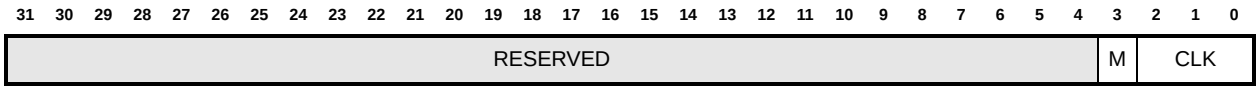
D

This bit is read-only.

This bit is set to 1 when JTAG_COM is written. It becomes 0 when the shifting has completed. Software can poll this bit to determine when a command has completed and it is therefore safe to retrieve the data in JTAG_TDO and to load new data into JTAG_TMS and JTAG_TDI.

39.5.3.6 JTAG Master Configuration, JTAG_CONFIG

FIGURE 39-9: JTAG_CONFIG



CLK

This field determines the clock rate of the JTAG_CLK signal. Options are shown in Table 39-4, "JTAG Clock Options":

TABLE 39-4: JTAG CLOCK OPTIONS

CLK Value	JTAG_CLK Clock Rate
0h	Reserved
1h	32.26MHz
2h	16.13MHz
3h	8.06MHz
4h	4.03MHz
5h	2.01MHz
6h	1.00 MHz
7h	500 KHz

APPLICATION NOTE: the ARC clock must be configured to be equal to or faster than the JTAG_CLK.

M

This bit controls Master/Slave JTAG multiplexing. When this bit is 0 (default), the JTAG port is configured as a slave. When this bit is 1, the JTAG port is configured as a Master.

39.6 JTAG Port Signal Interface Description

The signal pins are defined in Section 2.4.4, "JTAG Interface," on page 14.

The JTAG_CLK input is the clock that drives the JTAG interface. It is asynchronous to other clocks on-chip.

The TMS input is sampled on each rising edge of JTAG_CLK, and governs the transitions among the 16 states of the state machine (TAP) that controls the transfer of data.

The TDI input is the serial data input, shifted in during the Shift-IR and Shift-DR states of the TAP. It is sampled on rising edges of JTAG_CLK.

The TDO output is the serial data output. It is presented on falling edges of JTAG_CLK, 1/2 clock before each input shift, to provide setup and hold time to the next JTAG controller in the chain. The final TDO output pin, after all on-chip chaining (Figure 39-2) is held in high-impedance mode (floating) except when valid data is being presented. The enabled/disabled state of the pin is also changed on falling edges of JTAG_CLK.

The JTAG_RST# input is provides the Async JTAG RESET. Note that the reset state of the JTAG port is only local to the JTAG port: its effect is to keep the JTAG port in an idle state and to disengage it from the rest of the system, so that it does not affect other on-chip logic in this state.

39.7 Power, Clocks and Reset

See Section 41.11, "JTAG Interface Timing," on page 522 power on sequence and reset timing.

39.7.1 POWER DOMAINS

The JTAG block is powered by VTR.

39.7.2 CLOCKS

The JTAG port runs internally from the externally-provided JTAG_CLK clock pulses only. There is no requirement for JTAG_CLK to be constantly running.

The following JTAG Registers interface to the ARC Island block (as illustrated in Figure 39-2) for access to registers and memory: STATUS Register (8h), TRANSACTION COMMAND Register (9h), & ADDRESS Register (Ah), DATA Register (Bh). There is a clock relationship required between JTAG_CLK and the ARC Core clock frequency. JTAG_CLK may be asynchronous, but it must be slower than 1/2 the frequency of the ARC Core clock. In practical terms, then, JTAG_-CLK should be selected to be nominally 1/4 of the minimum Core clock frequency. See APPLICATION NOTE: on page 103.

APPLICATION NOTE: The Ashling JTAG interface box is not documented to operate any slower than JTAG_CLK = 1MHz, therefore the Core must be running at 4MHz in order to run the ARC debugger through the JTAG interface using the Ashling interface.

Stopping the Core clock disables the JTAG port for debugging purposes. It does not affect IDCODE or BYPASS operation.

See Section 41.11, "JTAG Interface Timing," on page 522 for the maximum frequency f_{clk} on the JTAG_CLK pin to access a JTAG Registers other than STATUS Register (8h), TRANSACTION COMMAND Register (9h), & ADDRESS Register (Ah), DATA Register (Bh).

39.7.3 RESET

The ARC JTAG block has two resets: Async JTAG RESET by its JTAG_RST# input and Sync JTAG RESET by JTAG protocol.

39.7.3.1 Async JTAG RESET

The JTAG_RST# pin provides the Async JTAG RESET to the JTAG Registers. The JTAG_RST# pin has an active low, asynchronous assertion and a synchronous de-assertion. The JTAG Registers will be reset asynchronously (and immediately) upon the active low JTAG_RST# assertion. Once the JTAG_RST# pin has been de-asserted, a delay of three JTAG_CLKs is required in order to access the JTAG Registers; i.e., the JTAG Registers will remain in reset for three clocks following the synchronous JTAG_RST# pin de-assertion. See Section 41.11, "JTAG Interface Timing," on page 522.

APPLICATION NOTE: After asserting and de-asserted the JTAG_RST# pin, a Sync JTAG RESET can be applied before starting to access the JTAG Registers (to meet the JTAG_RST# synchronous de-assertion requirement.

Note 39-1 JTAG registers, in particular the JTAG Test Mode Data Registers, are set to their initial values by the assertion of the JTAG_RST# pin, not the VTR Power On Reset. JTAG_RST# must be held low while the MEC1609/MEC1609i is powering up so the registers can be set to their proper default values. If JTAG_RST# is high during power up, the JTAG Test Mode Data Registers may be set to unpredictable values, which may trigger unwanted test modes.

APPLICATION NOTE: Care should be taken during VTR power up to insure that JTAG_RST# is asserted for a longer time then the VTR rise time due to capacitive loading. See Section 39.2.2, "Slave Select Timing," on page 477 for timing requirement.

39.7.3.2 Sync JTAG RESET

It can also be reset synchronously by a JTAG_CLK / TMS sequence, in accordance with the JTAG standard. A series of 5 successive JTAG_CLK rising edges, with TMS held high throughout, will accomplish this from any state.

The ARC JTAG port, upon entering its Reset state, will be prepared to accept an Instruction or Data transfer. It will also be disengaged from external circuitry, allowing it to operate normally.

The initial contents of the Instruction Register are the IDCODE command (Ch). If a Data transfer is performed first after Reset, without an preceding Instruction transfer, then the IDCODE value will be loaded into its 32-bit shift register and presented serially, after which will appear the bits shifted in from TDI.

The initial contents of the Data registers are as listed in Table on page 489.

39.8 Interrupts

There are no interrupts assigned to the ARC JTAG block. Control of the processor is performed by monitoring, setting and clearing the H bit (Halt) in the Aux register STATUS32, and by register manipulations while the processor is halted.

However, any interrupt or other event that can be triggered by accessing registers (Core, Aux or Memory-Mapped) can be triggered through the JTAG port.

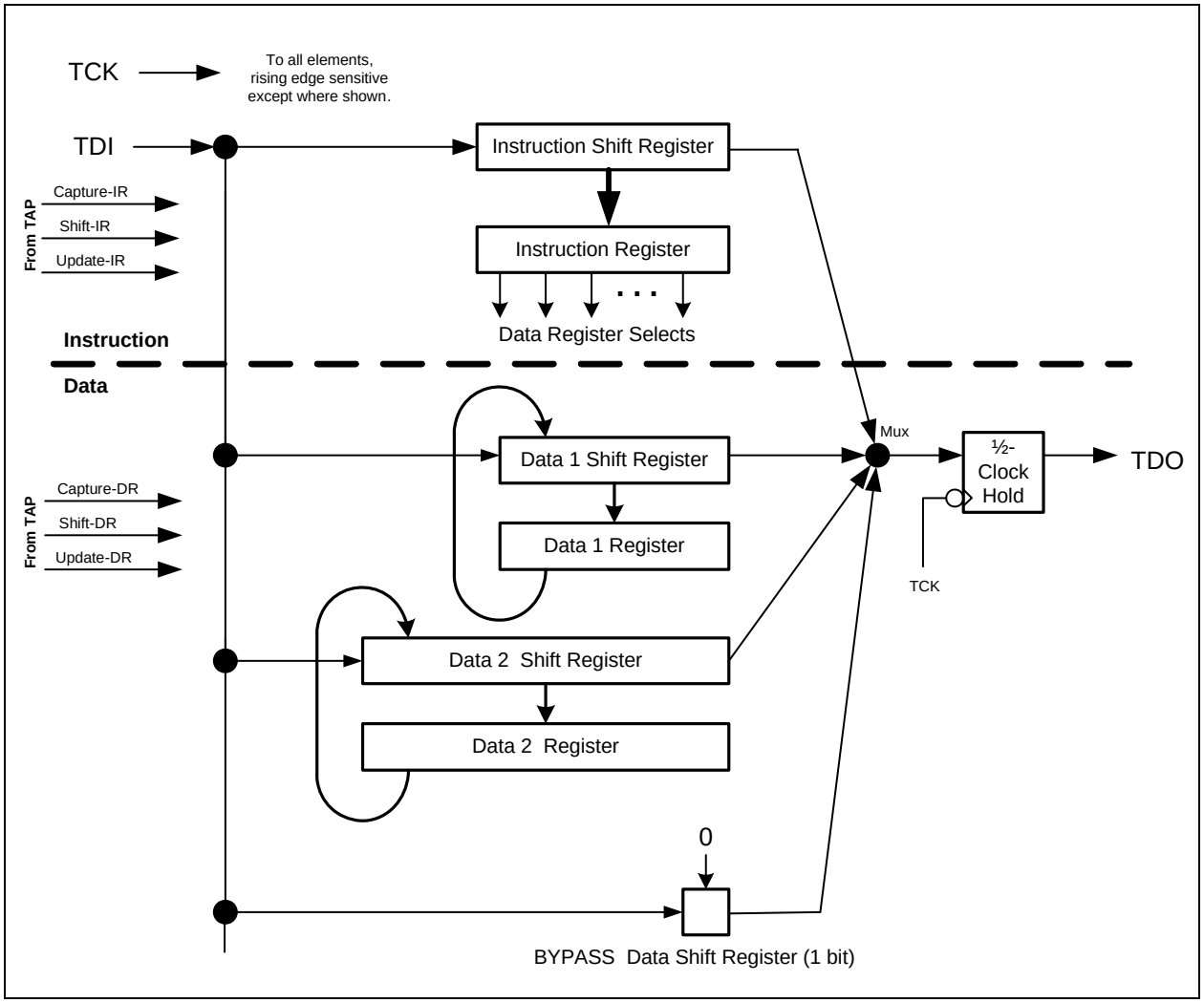
39.9 JTAG Background

The following is a simplified description, intended to provide background for the ARC JTAG port. For full details, see the JTAG specification (IEEE Standards 1149.1 and 1149.1b).

39.9.1 INTERNAL STRUCTURE

A JTAG port operates by transferring information serially into and out of an Instruction register and one or more Data registers. These registers are connected in parallel with each other, and can be of arbitrary length. See Figure 39-10.

FIGURE 39-10: STRUCTURE OF A JTAG PORT (SIMPLIFIED)



The protocol for shifting information makes a distinction between an Instruction transfer (to/from a single Instruction register) and a Data transfer (to/from one of several Data registers). The Instruction register is handled separately because it selects which specific Data register is accessed by subsequent Data transfers.

In daisy-chained JTAG controllers, the Instruction registers form one chain, and the currently-selected set of Data registers in each JTAG controller combine to form a second chain. To shorten the Data chain when not all JTAG controllers are of interest, a mandatory one-bit Data register called BYPASS is provided. There is no bypassing for the Instruction chain, so its full length must be shifted as each new instruction is transferred anywhere. Selecting the BYPASS Data register is the equivalent of a No-Operation instruction for a JTAG controller, and this instruction is always defined as a '1' in all Instruction register bits.

Each entity called a "Register" actually consists of two parts: the Register itself, and an associated Shift Register which connects to TDI and TDO. The Register may load from, and/or source information in parallel to, the Shift Register. These two parts are the same length, meaning that (for example) a 5-bit Register will be associated with a 5-bit Shift Register.

The Instruction register and the Data registers respond to decoded state signals from the TAP Controller sub-block (Section 39.9.2), which represent sub-steps of a transfer. The sub-steps they perform are Capture, which loads the shift register in parallel, Shift, which shifts information in from TDI and out on TDO, and Update, which writes information from the Shift Register in parallel. The Capture-IR, Shift-IR and Update-IR controls affect only the Instruction register. The Capture-DR, Shift-DR and Update-DR controls affect only the Data register that is currently selected by the contents of the Instruction register.

39.9.2 TAP CONTROLLER AND PROTOCOL

The JTAG protocol is driven by the level of the TMS (Test Mode Select) input pin at each rising edge of the JTAG_CLK clock. This is the responsibility of the TAP Controller section of the JTAG controller, which performs state transitions as illustrated in the state diagram in Figure 39-11. States whose names end with “IR” affect the Instruction register (the rightmost column of states in Figure 39-11), and those ending with “DR” affect a Data register (the middle column in Figure 39-11). Note that the TMS signal goes in parallel to all JTAG ports in a chain, so they are always in the same protocol state. The sequence of accessing any register is as follows:

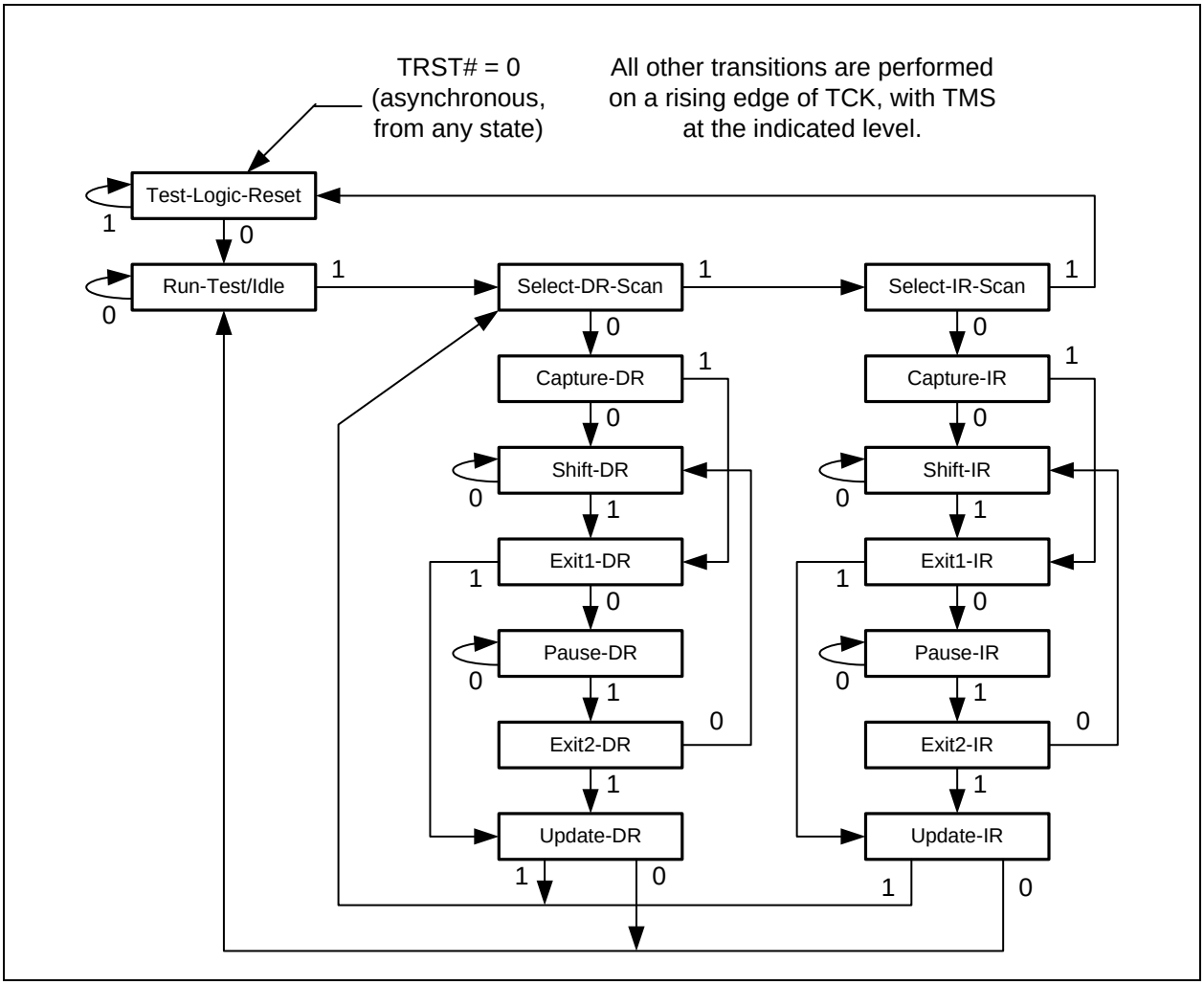
- Capture (IR or DR), which loads a shift register from its source in preparation for shifting it out. In the case of the Instruction register, this is a fixed value, and not the previous contents of the Instruction register. In the case of the BYPASS Data register, this is a fixed ‘0’ value. The Capture state is transitory, being present for only one JTAG_-CLK cycle, once per transfer.
- Shift (IR or DR), which shifts the Captured information in the Shift Register out on the TDO pin while also shifting information in from the TDI pin. The registers (by convention) shift from left to right, so the least-significant bit of a value is transferred first. This state may be held arbitrarily (holding TMS=0) to shift as many bits as desired.
- Update (IR or DR), which loads a Register from its Shift Register after the shifting has completed. The Update state is transitory, being present for only one JTAG_CLK cycle, once per transfer.

There is also a Pause state (IR or DR) which may be used to exit and re-enter the Shift state without terminating the transfer in progress. This state may be held (TMS=0) in order to delay for any desired number of JTAG_CLK cycles.

Outside of Instruction or Data transfers, there are two states which may be entered and held. These are shown in the leftmost column in Figure 39-11.

- The Test-Logic-Reset state holds the JTAG logic in its reset state. This re-initializes the registers that are internal to the JTAG logic. This state is entered asynchronously by assertion of JTAG_RST# low, and it can be seen in Figure 39-11 that, from any other state, this state will be entered by 5 successive JTAG_CLK cycles with TMS held to ‘1’.
- Run-Test/Idle holds JTAG logic idle, but not reset, between transfers.

FIGURE 39-11: TAP CONTROLLER STATE DIAGRAM



39.9.3 INTERFACE TIMING EXAMPLE

Figure 39-12 illustrates the timing relationship between data shifting and the TAP Controller's Shift states, using a 1-bit Data register as an example. (This is in fact the exact situation when the BYPASS Data register is selected: refer to FIGURE 39-10: on page 486.)

The TAP Controller changes states on each rising edge of JTAG_CLK, traversing the state table in Figure 39-11 as directed by the TMS input signal from the external interface.

Previous to the waveform in Figure 39-12, the TAP Controller has already passed through a Capture-DR state, so the 1-bit Shift Register has been pre-loaded with a "Capture Value", either from its associated parallel Register or from another source. (For the BYPASS register, this would be a fixed '0'.)

At the first rising edge of JTAG_CLK in Figure 39-12, the Shift-DR state is being entered. As yet, no valid data needs to be present on TDI or TDO.

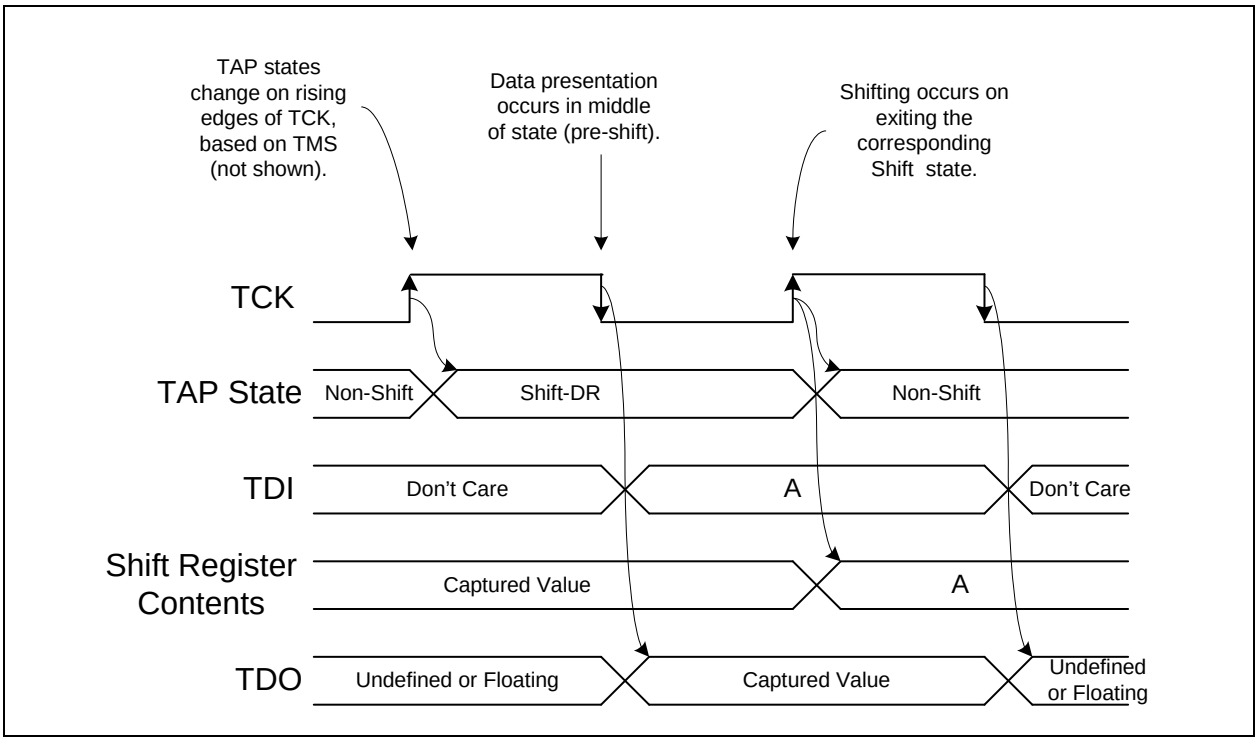
At the first falling edge of JTAG_CLK, while the Shift-DR state is active, the TDO pin begins presenting the least-significant bit of the Shift Register (the only bit, in this example), which is holding the Captured Value. At about this time also, the external interface will drive TDI to the desired new state for this Data register.

At the next rising edge of JTAG_CLK, the Shift-DR state is exited, and that same clock edge is used to actually perform the commanded shift. The TDI value "A" is shifted into the Shift Register. This same rising edge of JTAG_CLK is used by the external interface to shift in the Captured Value from TDO. The TDO output does not change yet, because it is held by a 1/2 clock delay stage (see FIGURE 39-10: on page 486), providing hold time for the external interface.

On the next falling edge, the TDO output changes. Since the Shift state is no longer present, TDO is not required at this time to present valid data, and in fact for an off-chip connection it is required to float at this time.

After this timing diagram completes, the TAP machine will continue to an Update-DR state, at which time the value A, now present in the Shift Register, will be written to its destination. (In the specific case of the BYPASS register, there is no destination, and that step will do nothing.)

FIGURE 39-12: TIMING ILLUSTRATION: 1-BIT DATA REGISTER



39.10 Registers

The ARC JTAG Port is defined in the ARC 600 External Interfaces Reference Manual, Chapter 2.

There are no JTAG registers accessible in any EC or AHB memory space. JTAG registers are accessible only through the JTAG pins themselves. (See Section 39.10.1, "Instruction Register," on page 489.)

39.10.1 INSTRUCTION REGISTER

The Instruction Register is four bits wide. It selects among the implemented Data Registers as listed in Table 39-5. When the Tap Controller is placed into the Test-Logic-Reset state, the Instruction register is initialized to Ch, selecting the IDCODE Data Register.

Registers marked as MCHP reserved must not be modified. Modifications may lead to unpredictable and unwanted behavior.

TABLE 39-5: ARC JTAG INSTRUCTION REGISTER ENCODINGS

Instruction Register Contents	Data Register Selected	Function of Data Register	Width (Bits)	State on JTAG Reset (Hex)
0h	(Reserved: EXTEST)	Not implemented, but reserved as required by JTAG standard.	32	0000_0000h
1h	(Reserved: SAMPLE/PRELOAD)	Not implemented, but reserved as required by JTAG standard.	32	0000_0000h
2h	RESET TEST	RESET TEST Register (2h)	32	0000_0000h

TABLE 39-5: ARC JTAG INSTRUCTION REGISTER ENCODINGS (CONTINUED)

Instruction Register Contents	Data Register Selected	Function of Data Register	Width (Bits)	State on JTAG Reset (Hex)
3h	TEST - MCHP Reserved		32	0000_0000h
4h	MCHP Reserved	Reserved for future use.)	32	0000_0000h
5h	(Reserved)	(Reserved for future use.)	32	0000_0000h
6h	(Reserved)	(Reserved for future use.)	32	0000_0000h
7h	(Reserved)	(Reserved for future use.)	32	0000_0000h
8h	STATUS	STATUS Register (8h) Status of Current Debugger Transaction (Read-Only)	4	undefined (based on bus status)
9h	TRANSACTION COMMAND	TRANSACTION COMMAND Register (9h) Initiates / Specifies a Debugger Transaction	4	3
Ah	ADDRESS	ADDRESS Register (Ah) Address of a Debugger Transaction	32	0000_0000h
Bh	DATA	DATA Register (Bh) Data In / Data Out for Debugger Transactions	32	Out = 0000_0000h In = undefined
Ch	IDCODE	IDCODE Register (Ch) JTAG Standard IDCODE Register (Capture = Read-Only fixed value)	32	1000_24B1h
Dh	MCHP Reserved		32	0000_0000h
Eh	(Reserved)	Reserved for future use.	32	0000_0000h
Fh	BYPASS	BYPASS Register (Fh) JTAG Standard BYPASS Register (Capture = Read-Only ‘0’)	1	0

39.10.2 JTAG DEBUG DATA REGISTERS

Note 39-1 Unfortunately, ARC names one of its JTAG Debug Data registers “DATA”. To avoid confusion, while maintaining the terminology in both ARC and JTAG documentation, the term “Data register” will refer to any of the JTAG Data registers, and the term “DATA register” (all upper-case) will refer to the specific JTAG Data register that is selected by Instruction Register = B. See Section 39.10.2.2, "DATA Register (Bh)," on page 491.

The Debug Data Register set of the ARC JTAG Port provide the means for an external JTAG-connected debugger system to monitor and control the execution of a program. Using the JTAG Data registers ADDRESS, DATA, TRANSACTION COMMAND and STATUS, the debugger can perform “transactions” to read or write:

- Any Aux Register, giving it the ability to start, halt or step a program, and alter the PC and/or program status
- Any addressable memory or I/O location, as an aligned 32-bit value
- Any Core Register, if the processor is in a halted state

To write to a specific register or a memory location, the debugger will place the desired register number or memory address into the ADDRESS register, place the value to be written into the DATA register, and then trigger the transfer by placing the direction and addressing space (Core register / Aux register / Memory) into the TRANSACTION COMMAND register. It will then read the STATUS register until it indicates that the transaction is finished.

To read from a specific register or memory location, the debugger will place the desired register number or memory address into the ADDRESS register, and trigger the transfer by placing the direction and addressing space (Core register / Aux register / Memory) into the TRANSACTION COMMAND register. It will then read the STATUS register until it indicates that the transaction is finished, and read the DATA register to access the value.

Optimizations are possible in repeated accesses, because of the actions of the ADDRESS and DATA registers, as described in Section 39.10.2.1 and Section 39.10.2.2.

39.10.2.1 ADDRESS Register (Ah)

The ADDRESS register is a 32-bit register which receives from the debugger either a Core Register number, an Aux Register number or an address in the Memory space (memory or I/O).

Note:	As a memory address, the low-order 2 bits of the ADDRESS register are ignored (assumed by hardware to be 00), and a full 32-bit value is referenced at that location. There is no way for the debugger to specify a smaller width of data, and so a write to a single byte (for example) is performed using a read transaction followed by a write transaction, preserving the values of the unaffected bytes.

After use, the ADDRESS register automatically increments, by 1 if a register was accessed, and by 4 if a memory location was accessed. Therefore, as long as the JTAG TAP Controller is not brought to the Test-Logic-Reset state between accesses, it is not necessary to provide a new ADDRESS register value between transactions involving successive registers or memory locations. (The Test-Logic-Reset state must be avoided because it resets the value of the ADDRESS register.)

TABLE 39-6: ADDRESS REGISTER

INSTRUCTION REGISTER CONTENTS	Ah			32 bits			**REGISTER SIZE**	
POWER	VTR			0000_0000h			Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**…**		**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	-	-	-	-	-	-	-	-
BIT NAME	Address[31:0]							

39.10.2.2 DATA Register (Bh)

The DATA register is a 32-bit register which is the ARC JTAG Port's portal for data values that are being read or written by a transaction. When writing to a register or memory, the DATA register will be set up by the debugger before the transaction is triggered. When reading from a register or memory, the DATA register will be read by the debugger as the last step of the transaction. See Note 39-1 on page 484.

The DATA register is not affected at the end of a write transaction, so (for example) to fill successive locations with the same value it is not necessary to provide it again, as long as the Test-Logic-Reset of the JTAG TAP Controller is not entered (which would clear it).

TABLE 39-7: DATA REGISTER

INSTRUCTION REGISTER CONTENTS	Bh			32 bits			**REGISTER SIZE**	
POWER	VTR			0000_0000h			Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**…**		**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	-	-	-	-	-	-	-	-
BIT NAME	Data[31:0]							

39.10.2.3 TRANSACTION COMMAND Register (9h)

The TRANSACTION COMMAND register is written by the debugger to trigger a transaction. It is a 4-bit register, which is written with one of the values in Table 39-9 to specify the direction and addressing space of the transaction.

TABLE 39-8: DATA REGISTER

INSTRUCTION REGISTER CONTENTS	9h		4 bits	**REGISTER SIZE**
POWER	VTR		0h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**
BIT	**BIT3**	**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	-	-	-	-
BIT NAME	Command[3:0]			

TABLE 39-9: TRANSACTION COMMAND REGISTER ENCODINGS

Encoding (Binary)	Transaction Type
0000	Write to Memory space
0001	Write to a Core register
0010	Write to an Aux register
0011	No Operation
0100	Read from Memory space
0101	Read from a Core register
0110	Read from an Aux register
0111	(obsolete Write form)
1000	(obsolete Read form)
(other)	Reserved

39.10.2.4 STATUS Register (8h)

The STATUS register is a 4-bit read-only register. It is read by the debugger to determine when a transaction has completed internally, and when the next transaction may be started. It also provides additional status information useful to the debugger.

TABLE 39-10: STATUS REGISTER

INSTRUCTION REGISTER CONTENTS	8h		4 bits	**REGISTER SIZE**
POWER	VTR		0h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**
BIT	**BIT3**	**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	R	R	R	R
BIT NAME	-PC	-RD	FL-	ST-

(ST): STALLED

1 = The current transaction is stalled (busy)

0 = The current transaction is not stalled (not busy)

(FL): FAILURE

1 = The transaction has failed

0 = The transaction has not failed

A transaction will fail if it attempts to access a Core register while the processor is running. Bus errors should also set this bit.

(RD): READY

1 = The transaction is finished (ready)

0 = The transaction is not finished

(PC): PC_SEL

This bit has no direct hardware effect. It displays the state of the PC_SEL signal, which is bit 0 of the write-only Aux register PCPORT (Aux Register #24h). This bit is initialized to '1' on a processor reset, and is used internally by the debugger system as a means to communicate configuration information.

39.10.3 JTAG STANDARD DATA REGISTERS

39.10.3.1 IDCODE Register (Ch)

This is a 32-bit read-only register containing the hex value 1000_24B1. It serves to identify the ARC JTAG Port as belonging to an ARC600 core, in a component containing one processor.

IDCODE registers are required to conform to the JTAG standard, and they contain an 11-bit Manufacturer ID number.

TABLE 39-11: IDCODE REGISTER

INSTRUCTION REGISTER CONTENTS	Ch			32 bits			**REGISTER SIZE**	
POWER	VTR			1000_24B1h			Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**…**		**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	R	R	R	R	R	R	R	R
BIT NAME	IDCODE[31:0]							

39.10.3.2 BYPASS Register (Fh)

The BYPASS register consists only of a 1-bit shift register cell. The Capture-DR state clears it to '0' when selected. The Update-DR state does nothing.

The function of this register is to provide the minimum amount of delay (one bit of '0') when other JTAG ports on the chain are being exercised.

TABLE 39-12: BYPASS REGISTER

INSTRUCTION REGISTER CONTENTS	Fh	1 bit	**REGISTER SIZE**
POWER	VTR	1000_24B1h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**
BIT	**BIT0**		
JTAG TYPE			
BIT NAME	BYPASS		

39.10.4 JTAG TEST MODE DATA REGISTERS

JTAG Test Registers are 32-bit read/write registers that are used for test functions. These registers are always available to the JTAG port, even if access to the other JTAG registers is blocked.

39.10.4.1 RESET TEST Register (2h)

The RESET TEST Register is a 32-bit register used to explicitly control reset functions inside the MEC1609/MEC1609i. The default for this register is 0000_0000h.

TABLE 39-13: RESET TEST REGISTER

INSTRUCTION REGISTER CONTENTS	2h				32 bits		**REGISTER SIZE**	
POWER	VTR				0000_0000h		Async JTAG RESET **DEFAULT**	
BIT	**Bits 31-14**		**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
JTAG TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved		Test	Test	Test	Test	Test	Test
BIT	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
JTAG TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Test	Test	Test	Test	POR EN	VTR POR	VCC POR	Test

ME

Mass Erase. If this bit is ‘1’ when the internal VTR Power On Reset signal transitions from ‘0’ to ‘1’, the Embedded Flash Subsystem will enter the Emergency Mass Erase mode, which will erase the entire Flash whether or not the Boot_-JTAG_Block bit in the Embedded Flash Initialization Register is set.

VCC POR

Asserts VCC Power On Reset: When the VCC POR active low bit is asserted ‘0’ while the field POR EN in this register is ‘1’, forces a VCC Power On Reset. When the VCC POR active low bit de-asserted ‘1’, the VCC POR circuitry returns to its normal state.

VTR POR

Asserts VTR Power On Reset: When the VTR POR active low bit is asserted ‘0’ while the field POR EN in this register is ‘1’, forces a VTR Power On Reset. When the VTR POR active low bit de-asserted ‘1’, the VCC POR circuitry returns to its normal state.

POR EN

Power On Reset Enable. When ‘1’, the reset functions controlled by VCC POR and VTR POR are enabled. When ‘0’, the VCC POR and VTR POR fields in this register have no effect on the POR circuitry.

TEST

All TEST bits should be set to ‘0’ when writing this register.

39.10.4.2 TEST REGISTER 4/Reset Register (Dh)

The RESET TEST Register is a 32-bit register used to explicitly control reset functions inside the MEC1609/MEC1609i. The default for this register is 0000_0000h.

TABLE 39-14: TEST REGISTER 4/RESET REGISTER

INSTRUCTION REGISTER CONTENTS	Dh			32 bits			**REGISTER SIZE**	
POWER	VTR			0000_0000h			Async JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**BIT28**	**BIT27**	**BIT26**	**BIT25**	**BIT24**
JTAG TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**BIT23**	**BIT22**	**BIT21**	**BIT20**	**BIT19**	**BIT18**	**BIT17**	**BIT16**
JTAG TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	src_clk stop_n	jtag halt override	Test regfile bypass_en	Test regfile_ select	Test test_rosc_ en	Test test_clk_ req_en	Test ring enable en	Test ring_coast en
BIT	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
JTAG TYPE	R/W	R/W	R	R	R/W	R/W	R/W	R/W
BIT NAME	Test ring_en override	Test ring_ coast override	Test ring_coast state	Test clk_req_ state	Test TROE	Test TROS		
BIT	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
JTAG TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Test arc_clk_di sable	Test wake	Test ring_en	Test test_pulls	Test test_vohv ol	Test test_drive _en	Test_ XNOR_ En	ARC_ Fast_ Reset

ARC_FAST_RESET

If this bit is ‘1b’, the reset going to the ARC processor and select peripherals is reduced from its nominal 20ms duration. If this bit is ‘0b’, the ARC reset is stretched by the nominal delay.

TEST_XNOR_EN

If this bit is ‘1b’, the Device-Under-Test XNOR chain test mode is enabled. If this bit is ‘0b’, the XNOR mode is disabled. See Section 39.11, "XNOR Chain," on page 496.

Note:	Once the XNOR chain is enabled, a power cycle is required to re-establish JTAG operation or.

TEST

All TEST bits should be set to ‘0; when writing this register.

39.10.5 JTAG STANDARD PORT DISCOVERY

This section provides information that is not unique to ARC, but is part of the JTAG standard, and is provided for information.

The Discovery process will identify each JTAG controller that has an IDCODE register. Part of what needs to be derived is the length of the Instruction register in each of the JTAG ports. If this cannot be derived from the IDCODE values, or if some JTAG ports do not have an IDCODE register, then the missing lengths must be provided by other means.

In the Test-Logic-Reset state, a JTAG port is required to initialize its Instruction register to select the IDCODE Data register if present, or if it is not present, then to select the BYPASS Data register.

The IDCODE Data register:

- Must be exactly 32 bits in length
- Must have '1' in its first (least-significant) bit
- Must not have the pattern 000011111111 (FFh) in its first (least-significant) 12 bits.
- Will contain a completely definitive port identification, because 11 bits of it are a Manufacturer ID number assigned by the JEDEC standards organization.

A BYPASS Data register access will initialize its 1-bit shift register to '0' at the Capture-DR state, effectively making the BYPASS register appear to be 1-bit read-only '0'.

Discovery, therefore, consists of the external JTAG host doing the following:

- Place the chain of JTAG controllers into the Test-Logic-Reset state.
- Do a Data register access, without an Instruction register access first.
  - This Data access will shift in 8 bits of ones, followed by all zeroes for the duration of the discovery phase.
- While shifting, examine the data appearing on TDO for IDCODE values.
  - A '0' indicates a JTAG port that has no IDCODE register. Collect only this bit, and note that the JTAG port exists. Start looking for an IDCODE value at the next bit.
  - A '1' indicates that an IDCODE register is coming. Collect this bit and the next 31 bits to identify the JTAG port. If, however, the value seen is 00h0000FF, then this is ensured to be the value provided originally on TDI, and indicates the end of the chain.

39.11 XNOR Chain

39.11.1 OVERVIEW

The XNOR Chain test mode allows users to confirm that all MEC1609/MEC1609i pins are in contact with the motherboard during assembly and test operations. The XNOR Chain test mode is enabled and disabled through the JTAG interface, using bit Test_XNOR_En in JTAG TEST REGISTER 4/Reset Register (Dh).

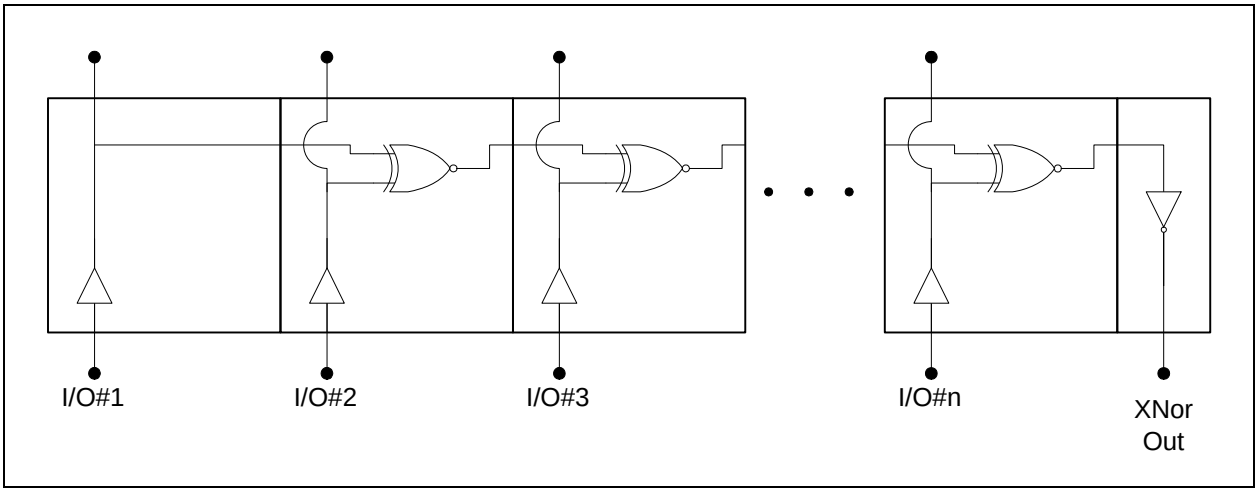
An example of an XNOR Chain test structure is illustrated below in Figure 39-13. When the XNOR Chain test mode is enabled all pins except for the Excluded Pins shown in Section 39.11.2 are disconnected from their internal functions and forced as inputs to the XNOR Chain. This allows a single input pin to toggle the XNOR Chain output if all other input pins are held high or low. The XNOR Chain output is the nRESET_OUT pin.

The tests that are performed when the XNOR Chain test mode is enabled require the board-level test hardware to control the device pins and observe the results at the XNOR Chain output pin; e.g., as described in Section 39.11.3, "Test Procedure," on page 497.

39.11.2 EXCLUDED PINS

The following pins are XNOR Chain Excluded Pins: POWER PLANE pins, VR_CAP, nRESET_OUT, XTAL1, XTAL2 and JTAG_RST#. (See Section 2.4.4, "JTAG Interface," on page 14.)

FIGURE 39-13: XNOR CHAIN TEST STRUCTURE



39.11.3 TEST PROCEDURE

39.11.3.1 Setup

1. Connect the VSS and AGND pins to ground.
2. Connect the VBAT and VTR pins to an unpowered 3.3V power source.
3. Connect an oscilloscope or voltmeter to the nRESET_OUT pin.
4. All other pins should be tied to ground.

Warning: Ensure power supply is off during Setup.

39.11.3.2 Testing

1. Turn on the 3.3V power source.
2. Enable the XNOR Chain through the JTAG interface (Test_XNOR_En in JTAG TEST REGISTER 4/Reset Register (Dh)). Note that at this point all inputs to the XNOR Chain are low and the output on the nRESET_OUT pin is high (refer to the Initial Configuration row in Table 39-15, "Toggling Inputs in Descending Pin Order").
3. Bring the highest numbered pin (N) high, where N is the number of pins to be tested as described in Note 39-2. The output on the nRESET_OUT pin should toggle (refer to Step 1 in Table 39-15.
4. In descending pin order successively bring each input high. As shown in Table 39-15 the nRESET_OUT pin toggles after each step. Continue until all inputs are high. The output on the nRESET_OUT pin is high (refer to the Final Configuration in Table 39-15).
5. The current state of the chip is now represented by the Initial Configuration row in Table 39-16, "Toggling Inputs in Ascending Pin Order".
6. Each input should now be brought low, starting at pin one (Step N+1) and continuing in ascending pin order until all inputs are low. The output on the nRESET_OUT pin is high (refer to the Final Configuration in Table 39-16.
7. Exit the XNOR Chain Test Mode by cycling VTR power.

TABLE 39-15: TOGGLING INPUTS IN DESCENDING PIN ORDER

	Pin Number (Note 39-2)							nRESET_OUT
	N	N - 1	N - 2	N - 3	N - 4	...	1	
Initial Configuration	L	L	L	L	L	L	L	H
Step 1	H	L	L	L	L	L	L	L
Step 2	H	H	L	L	L	L	L	H
Step 3	H	H	H	L	L	L	L	L
Step 4	H	H	H	H	L	L	L	H
Step 5	H	H	H	H	H	L	L	L
...	H	H	H	H	H	…	L	…
Step N-1	H	H	H	H	H	H	L	L
Final Configuration	H	H	H	H	H	H	H	H

TABLE 39-16: TOGGLING INPUTS IN ASCENDING PIN ORDER

	Pin Number (Note 39-2)							nRESET_OUT
	1	2	3	4	5	...	N	
Initial Configuration	H	H	H	H	H	H	H	H
Step N+1	L	H	H	H	H	H	H	L
Step N+2	L	L	H	H	H	H	H	H
Step N+3	L	L	L	H	H	H	H	L
Step N+4	L	L	L	L	H	H	H	H
Step N+5	L	L	L	L	L	H	H	L
...	L	L	L	L	L	…	H	…
Step N+(N-1)	L	L	L	L	L	L	H	L
Final Configuration	L	L	L	L	L	L	L	H

Note 39-2 pin numbers in these tables represent the number of pins to be tested and do not include the pins listed in Section 39.11.2, "Excluded Pins," on page 497.

40.0 ELECTRICAL SPECIFICATIONS

40.1 Maximum Ratings*

Operating Temperature Range (Commercial) ..... 0^o C to $+70^o$ C

Operating Temperature Range (Industrial)..... -40^o C to $+85^o$ C

Storage Temperature Range..... -55^o to $+150^o$ C

Lead Temperature Range .....Refer to JEDEC Spec J-STD-020B

Positive Voltage on any pin, with respect to Ground .....+5.5V

Negative Voltage on any pin, with respect to Ground ..... -0.3V

Supply Voltage Range VTR, VBAT, AVTR_ADC, VTR_REG, VTR_FLASH .....4.0 VDC

*Stresses above those listed above could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

TABLE 40-1: OPERATING CONDITIONS

Symbol	Parameter	MIN	TYP	MAX	Units
VBAT	Battery Backup Supply (TYPE 1)	2.97	3.3	3.63	V
	Battery Backup Supply (TYPE 2)	2.0	3.0	3.3	V
VTR, AVTR_ADC, VTR_REG, VTR_FLASH	Main Supply	2.97	3.3	3.63	V
PCI_CLK	PCI Clock		33		MHz
XTAL1/XTAL2	Clock Generator Crystal		32.768		KHz
T_A	Operating Temperature (Commercial)	0	–	70	°C
	Operating Temperature (Industrial)	-40	–	85	
V_{IH}	5V Tolerant Inputs	–	–	5.5	V
	Non 5V Tolerant Inputs	–	–	3.6	V
	Backdrive Protected Inputs (see Section 40.3, "Backdrive Protection," on page 503)	–	–	5.5	V

40.2 DC Specifications

40.2.1 ELECTRICAL CHARACTERISTICS

TABLE 40-2: DC ELECTRICAL CHARACTERISTICS

(T_A = 0° C - 70° C (Commercial), T_A = -40° C - 85° C (Industrial), VTR = 3.3 VDC ±10%)

All buffer types are 5V tolerant, except as indicated in Note 40-1.

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
I Type Input Buffer						TTL Levels
Low Input Level	V_{ILI}			0.8	V	
High Input Level	V_{IH}	2.0			V	
Input Leakage	I_{IL}	-10		+10	uA	V_{IN} = 0 to VTR
IS Type Input Buffer						
Low Input Level	V_{ILIS}			0.8	V	Schmitt Trigger
High Input Level	V_{IHIS}	2.2			V	Schmitt Trigger
Schmitt Trigger Hysteresis	V_{HYS}		250		mV	
Input Leakage	$I_{I\text{-}Leak'}$	-5		+5	uA	V_{IN} = 0 to VTR
Crystal Oscillator (XTAL2)						
OCLK	The MEC1609/MEC1609i crystal oscillator design requires a 32.768 KHz parallel resonant 12.5 pF load capacitance crystal with two 22 pF load caps. Refer to Application Note 19.3 PCB Layout Guide for MEC1609/MEC1609i for more information.					
ICLK						t_{RISE}/t_{FALL} = 1 µs max. There is no hysteresis on this input.
Low Input Level	V_{ILI}			0.8	V	
High Input Level	V_{IH}	2.0			V	
Input Leakage	I_{IL}	-10		+10	uA	V_{IN} = 0 to VTR
O8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -4 mA
Output Leakage	I_{OL}	-10		+10	µA	VIN = 0 to VTR
OD8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	V_{OL} = 8 mA
Output Leakage	I_{OH}	-10		+10	µA	I_{OH} = 0 to VTR
O12 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -6mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
OD12 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
OD16 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 16mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR

TABLE 40-2: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

(T_A = 0° C - 70° C (Commercial), T_A = -40° C - 85° C (Industrial), VTR = 3.3 VDC ±10%)

All buffer types are 5V tolerant, except as indicated in Note 40-1.

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
IO8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -4mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
IOD8 Type Buffer						
Low Output Level	V_{OL}			0.5	V	I_{OL} = 8 mA
High Input Level	V_{IH}	2.0			V	
Low Input Level	V_{IL}			0.8	V	
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
IO12 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -6mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
IOD12 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
High Input Level	V_{IH}	2.0			V	
Low Input Level	V_{IL}			0.8	V	
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
IO16 Type Buffer						
Low Output Level	V_{OL}			0.5	V	I_{OL} = 16mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -8mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
IOD16 Type Buffer						
Low Output Level	V_{OL}			0.5	V	I_{OL}=16 mA
High Input Level	V_{IH}	2.0			V	
Low Input Level	V_{IL}			0.8	V	
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR
IO24 Type Buffer						
Low Output Level	V_{OL}			.5	V	I_{OL} = 24mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -24mA
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to VTR

TABLE 40-2: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

(T_A = 0° C - 70° C (Commercial), T_A = -40° C - 85° C (Industrial), VTR = 3.3 VDC ±10%)

All buffer types are 5V tolerant, except as indicated in Note 40-1.

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PCI Buffers (PCI_ICLK, PCI_IO, PCI_I, PCI_O, PCI_OD)	V_{IH}	$0.5V_{CC}$		V_{CC} + 0.5	V	See *PCI Local Bus Specification Rev. 2.2* See Note 40-2
	V_{IL}	-0.5		$0.3V_{CC}$	V	
	V_{CC}	3.0		3.6	V	LPC Supply Voltage
	I_{IL}	-10		+10	µA	$0 < V_{IN} < V_{CC}$
	V_{OH}	$0.9V_{CC}$			V	I_{OUT} = -500 µA
	V_{OL}			$0.1V_{CC}$	V	I_{OUT} = 1500 µA
	C_{IN}			10	pF	
PECI (PECI_DAT, PECI_RDY)						All input and output voltages are a funciton of VREF_PECI buffer input.
Input voltage range	VIN	-0.3		VREF + 0.3	V	
Hysteresis	VHYS	0.1 * VREF	0.2 * VREF		V	
Low Input VLevel	VIL			0.275 * VREF	V	
High Input Level	VIH	0.725 * VREF			V	
Low Output Level	VOL			0.25 * VREF	V	IOL = 1 mA
High Output Level	VOH	0.75 * VREF			V	IOH = -6 mA
SB-TSI (SB-TSI_CLK, SB-TSI_DAT)						All input and output voltages are a funciton of VREF_PECI buffer input.
Input voltage range	VIN	-0.3		VREF + 0.3	V	
Hysteresis	VHYS	0.1 * VREF	0.2 * VREF		V	
Low Input VLevel	VIL			0.275 * VREF	V	
High Input Level	VIH	0.725 * VREF			V	
Low Output Level	VOL			0.25 * VREF	V	IOL = 1 mA
VREF_PECI						Connects to VTT and is Processor dependent.
Input Voltage	VI	0.95		1.26	V	
Input current	IDC			100	µA	
Input Leakage	ILEAK	-10		+10	µA	
Pull-Down Impedance	PD	65	91	136	K Ohms	
Pull-Up Impedance	PU	53	74	110	K Ohms	

Note 40-1 Non 5V Tolerant pins are listed in Table 2-5, “Non 5 Volt Tolerant Pins,” on page 12.

Note 40-2 All Signal functions in Section 2.5, "Pin Multiplexing," on page 21 listed as having an “VCC” emulated power tri-stated when the input VCC_PWRGD is inactive.

Note 40-3 All 5V Tolerant I-type & I/O-type input buffers can be pulled to 5 volts.

Note 40-4 All 5V Tolerant OD-type output buffers can be pulled to 5 volts.

Note 40-5 All 5V Tolerant O-type and I/O-type output buffers will only drive to 3.3 volts, even if pulled-up externally to 5 volts.

40.3 Backdrive Protection

All MEC1609/MEC1609i pins are backdrive protected (Table 40-3) with the exception of the Analog Data Acquisition Interface, the XTAL 1 and XTAL 2 pins, and the LPC Bus Interface. The pins in the Analog Data Acquisition Interface that share a GPIO and ADC signal function (see Section 2.4.6, Table 2-28, Table 2-29 and Table 2-30) are not backdrive protected regardless of the pin function and backdrive the AVTR_ADC pin.

TABLE 40-3: Backdrive Protection

(T_A = 0° C - 70° C (Commercial), T_A = -40° C - 85° C (Industrial))

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Input Leakage	I_{IL}	-10		+10	µA	V_{IN} = 5.0 V (5.5 V) @ VTR = 0 V (Note 40-6)

Note 40-6 for VBAT powered pins, VTR = 0 V, VBAT = 0 V.

40.4 Power Consumption

TABLE 40-4: MEC1609/MEC1609I POWER CONSUMPTION

VCC2 (V_{CC})	VCC1 (v_{TR})	System "S" State	ARC State	Clock State	Supply Current				Comments
						TYP (25^0C)	MAX (70^0C)	MAX (85^0C)	
3.3V	3.3V	S0–S2	Run	Ring OSC @ 64 MHz	ICC1	25 mA	28 mA	29 mA	FULL POWER f = 21.3 MHz (Note 40-7)
						20 mA	23 mA	24 mA	FULL POWER f = 8 MHz (Note 40-7)
		S0–S2	Sleep		ICC1	14 mA	17 mA	18 mA	EC SLEEP All Master Clock Trees except EC CLOCK TREE 0 are 'on.'
		S0–S2	Sleep		ICC1	7.5 mA	11 mA	12 mA	SYSTEM LIGHT SLEEP One Master Clock Trees is 'on.'
		S0–S2	Sleep		ICC1	1.5 mA	3 mA	3.5 mA	Clock-tree Gating in Heavy Sleep (see Section 5.4.7.3, "Clock-tree Gating in Heavy Sleep," on page 85).
0V		S3	Run		ICC1	24 mA	26 mA	27 mA	f = 21.3 MHz (Note 40-8)
						20 mA	22 mA	23 mA	f = 8 MHz (Note 40-8)
		S3*	Sleep		ICC1	1.5 mA	3 mA	3.5 mA	Clock-tree Gating in Heavy Sleep (see Section 5.4.7.3, "Clock-tree Gating in Heavy Sleep," on page 85
		S3*	Sleep	Stop	ICC1	450 µA	850 µA	950 µA	Note 40-10, Note 40-11
	0V	S5	Off	External 32KHz Clock Input	ICC0	2 µA	6 µA @ 25 °C		2.0V < V_{bat} < 3.0V XOSEL = 1
				Local 32KHz Crystal	ICC0	3 µA	9 µA @ 25 °C		2.0V < V_{bat} < 3.0V XOSEL = 0
				32KHz Disabled	ICC0	1 µA	3 µA @ 25 °C		2.0V < V_{bat} < 3.0V XOSEL = X

* Notes: On AC power, System can enter the S3-S5 states when ARC is in sleep mode.
ICC0 is V_{BAT} current; ICC1 is VTR current.

Note: All inputs not being tested are pulled up to power rails; all outputs are floating.

Note 40-7 All Master Clock Trees are running; one block is enabled in each clock tree as follows: EC CLOCK TREE 0: GPIO; EC CLOCK TREE 1: 16-Bit Counter/Timer; EC CLOCK TREE 2: PWM0; EC CLOCK TREE 3: PECI; HOST CLOCK TREE 0: LPC; HOST CLOCK TREE 1: UART.

Note 40-8 All EC Master Clock Trees are running as described in Note 40-7; HOST CLOCK TREE 0 and HOST CLOCK TREE 1 are 'off.'

Note 40-9 the MEC1609/MEC1609i Power Management (PM) States are described in Section 5.4.7.2, "EC Controlled Dynamic Power States," on page 84. The XOSEL bit in located in the Clock Enable Register on page 111.

Note 40-10 To achieve the lowest possible suspend current the VREF_PECI pin must be grounded even if the VREF_PECI function is not selected using the GPIO Interface Pin Control Register.

Note 40-11 to achieve the lowest possible suspend current the VREG Power-down Mode bit must be asserted (see Test Register 2Fh, Bit 7 in Table 4-19, “Chip-Level (Global) Control/Configuration Registers,” on page 71.

40.5 AC Specifications

AC Test Conditions

CAPACITANCE T_A = 25°C; fc = 1MHz; V_{cc} = 3.3 VDC

Parameter	Symbol	Limits			Units	Test Condition
		MIN	TYP	MAX		
Clock Input Capacitance	C_{IN}			20	pF	All pins except pin under test tied to AC ground
Input Capacitance	C_{IN}			10	pF	
Output Capacitance	C_{OUT}			20	pF	

41.0 TIMING DIAGRAMS

41.1 VTR/VBAT Power-up and Power-down Timing

FIGURE 41-1: VTR/VBAT POWER-UP

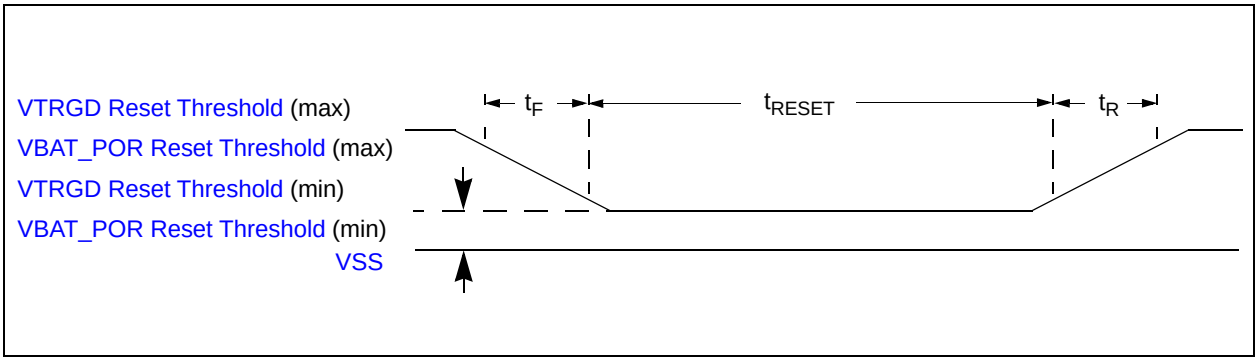


TABLE 41-1: VTR/VBAT POWER-UP TIMING PARAMETERS

Sumbol	Parameter	Limits		Units	Comments
		MIN	MAX		
t_F	VTR/VBAT Fall time	30		µs	
t_R	VTR/VBAT Rise time	0.150	30	ms	
t_{RESET}	Minimum Reset Time	1		µs	

41.2 Clock and Reset Timing

FIGURE 41-2: PCI CLOCK TIMING

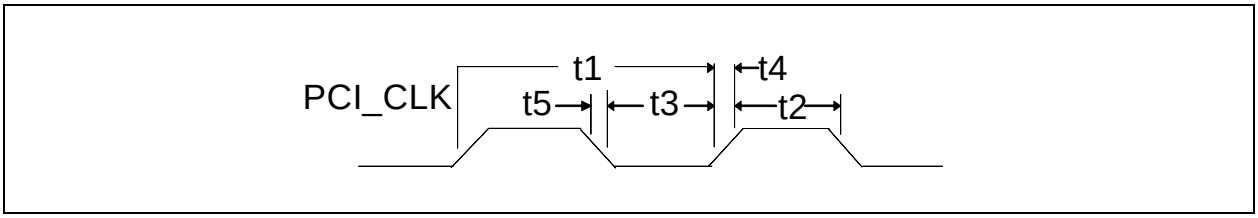


TABLE 41-2: PCI CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	30		33.3	nsec
t2	High Time	11			
t3	Low Time				
t4	Rise Time			3	
t5	Fall Time				

FIGURE 41-3: RESET TIMING

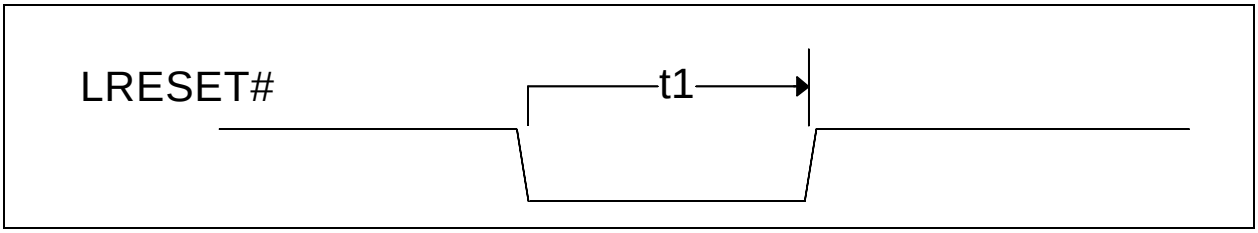


TABLE 41-3: RESET TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	LRESET# width	1			ms

41.3 LPC Timing

FIGURE 41-4: OUTPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS

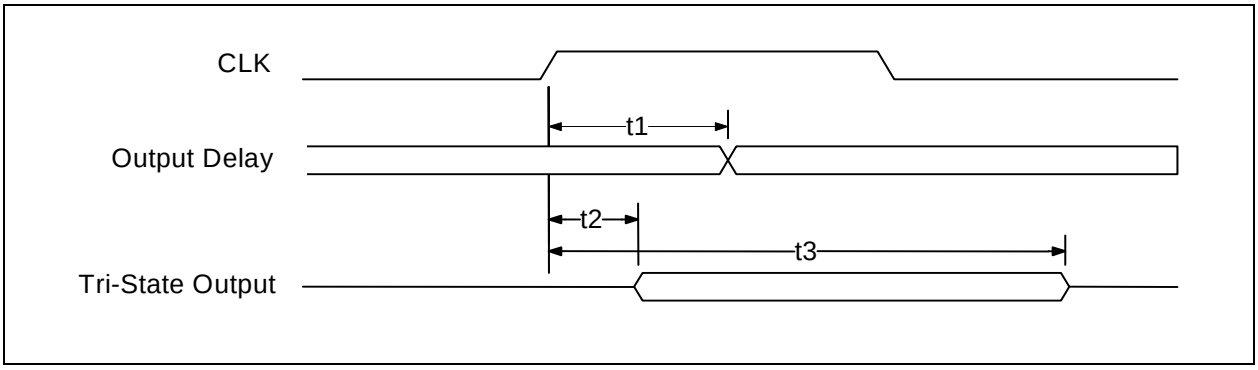


TABLE 41-4: OUTPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	CLK to Signal Valid Delay – Bused Signals	2		11	ns
t2	Float to Active Delay				
t3	Active to Float Delay			28	

FIGURE 41-5: INPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS

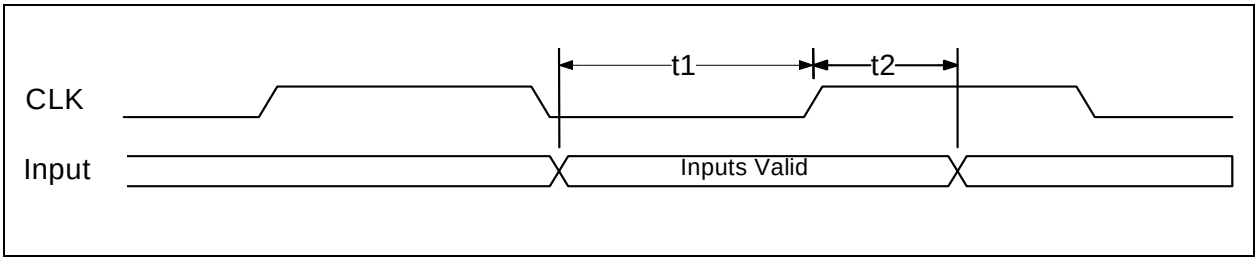
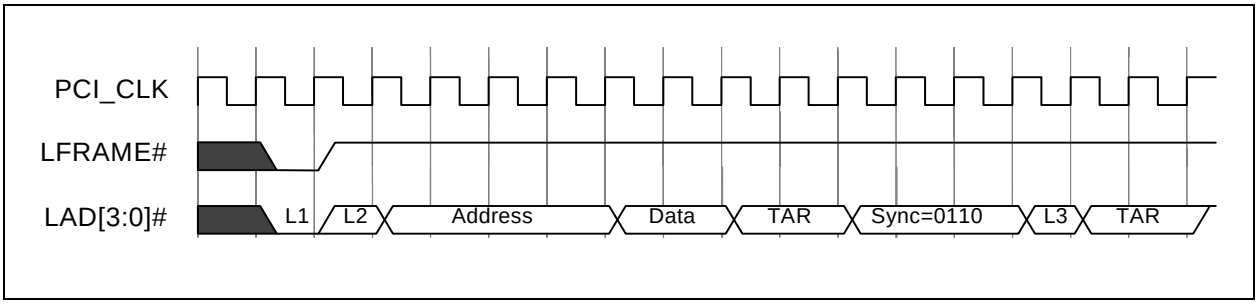


TABLE 41-5: INPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS PARAMETERS

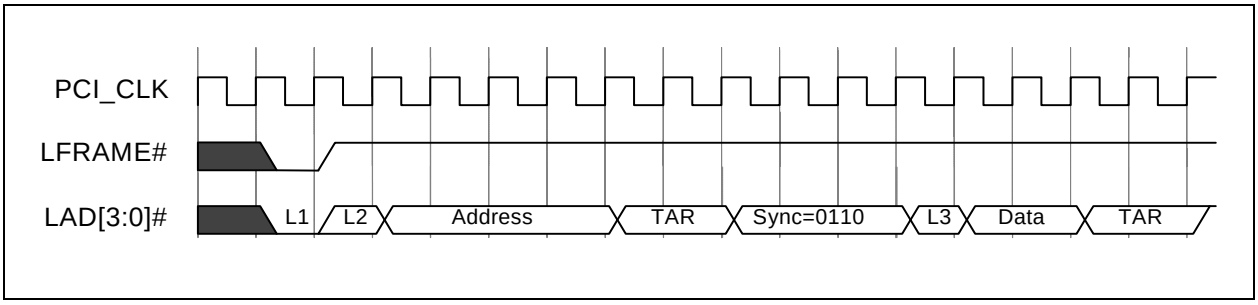
Name	Description	MIN	TYP	MAX	Units
t1	Input Set Up Time to CLK – Bused Signals	7			ns
t2	Input Hold Time from CLK	0			

FIGURE 41-6: I/O WRITE



Note: L1=Start; L2=CYCTYP+DIR; L3=Sync of 0000

FIGURE 41-7: I/O READ



Note: L1=Start; L2=CYCTYP+DIR; L3=Sync of 0000

FIGURE 41-8: DMA REQUEST ASSERTION THROUGH LDRQ#

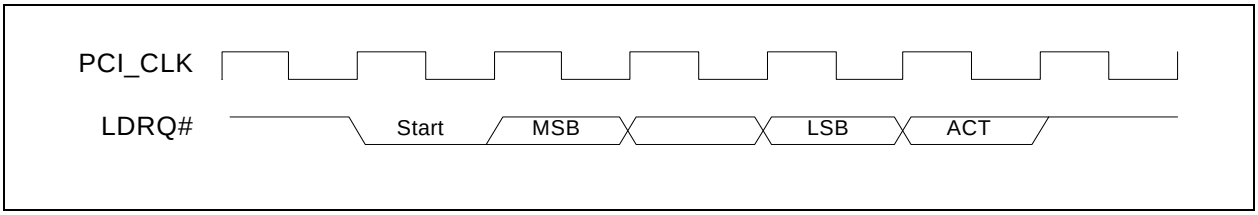
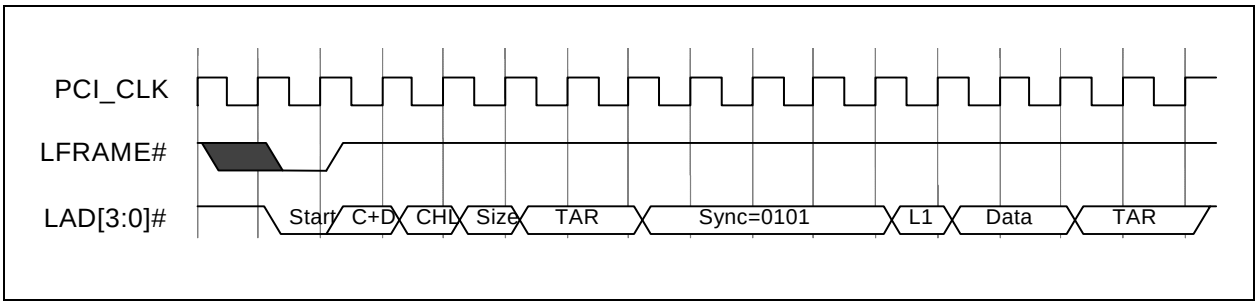
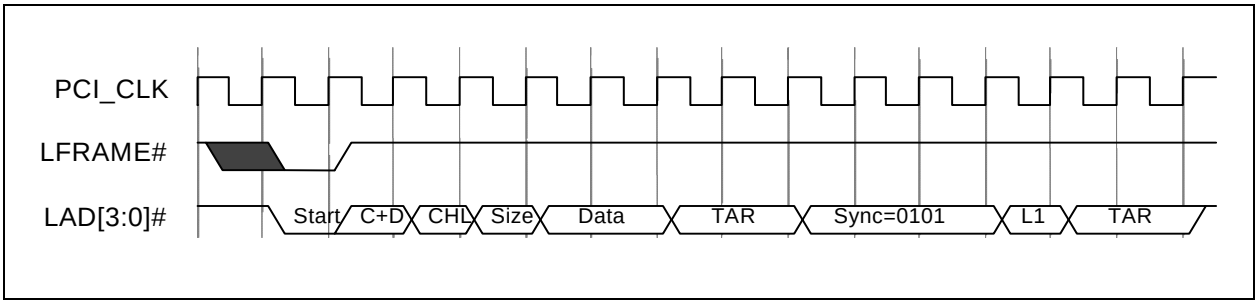


FIGURE 41-9: DMA WRITE (FIRST BYTE)



Note: L1=Sync of 0000

FIGURE 41-10: DMA READ (FIRST BYTE)



Note: L1=Sync of 0000

41.4 Serial IRQ Timing

FIGURE 41-11: SETUP AND HOLD TIME

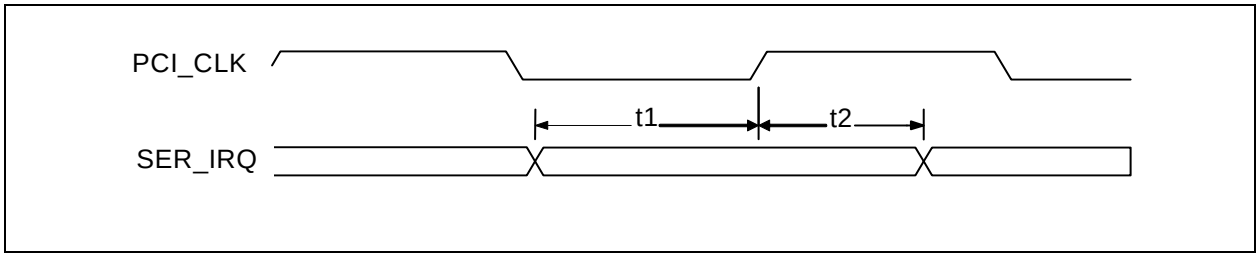


TABLE 41-6: SETUP AND HOLD TIME

Name	Description	MIN	TYP	MAX	Units
t1	SER_IRQ Setup Time to PCI_CLK Rising	7			nsec
t2	SER_IRQ Hold Time to PCI_CLK Rising	0			

41.5 Serial Port Data Timing

FIGURE 41-12: SERIAL PORT DATA

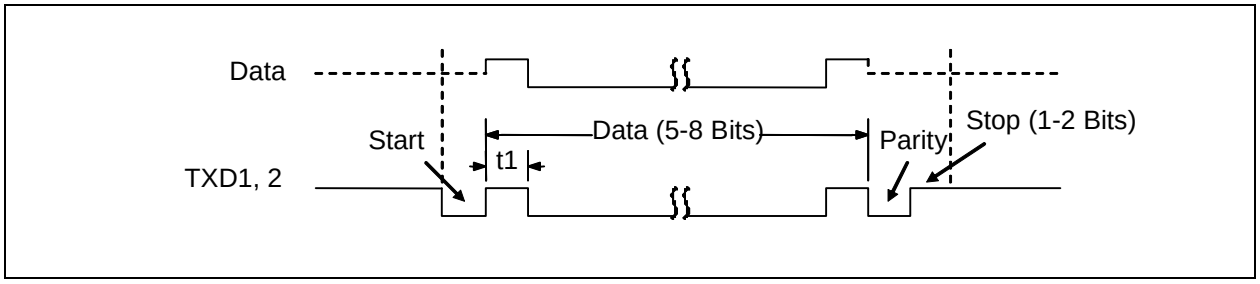


TABLE 41-7: SERIAL PORT DATA PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Serial Port Data Bit Time		t_{BR} (Note 4 1-1)		nsec

Note 41-1 t_{BR} is 1/Baud Rate. The Baud Rate is programmed through the divisor latch registers. Baud Rates have percentage errors indicated in Table 13-21, “UART Baud Rates (1.8432MHz source),” on page 216.

FIGURE 41-13: UART_CLK EXTERNAL CLOCK TIMING

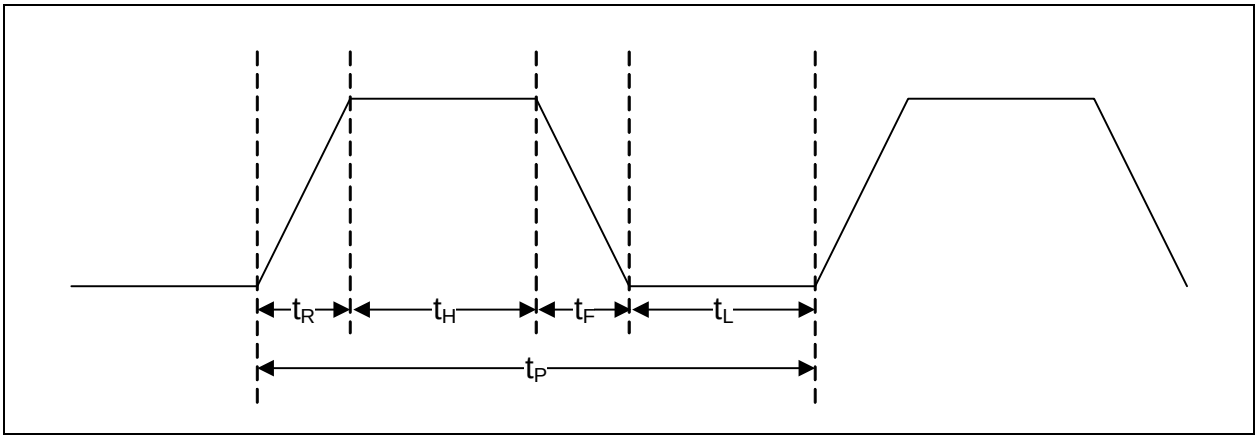


TABLE 41-8: UART_CLK EXTERNAL CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_P	Period	553.6	542.5	553.6	nsec
t_H	High Time	200			
t_L	Low Time				
t_R	Rise Time			10	
t_F	Fall Time				

41.6 I2C/SMBus Timing

FIGURE 41-14: I2C/SMBUS TIMING

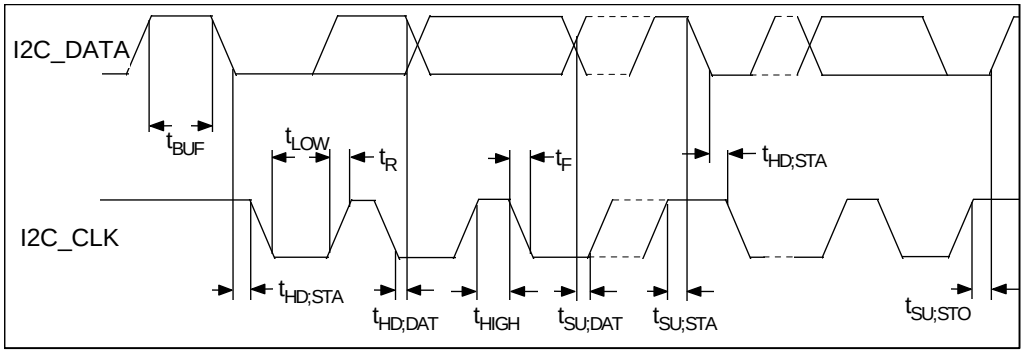


TABLE 41-9: I2C/SMBUS TIMING PARAMETERS

Symbol	Parameter	Standard-Mode		Fast-Mode		Units
		MIN	MAX	MIN	MAX	
f_{SCL}	SCL Clock Frequency		100		400	KHz
t_{BUF}	Bus Free Time	4.7		1.3		µs
$t_{SU;STA}$	START Condition Set-Up Time	4.7		0.6		µs
$t_{HD;STA}$	START Condition Hold Time	4.0		0.6		µs
t_{LOW}	SCL LOW Time	4.7		1.3		µs
t_{HIGH}	SCL HIGH Time	4.0		0.6		µs
t_R	SCL and SDA Rise Time		1.0		0.3	µs
t_F	SCL and SDA Fall Time		0.3		0.3	µs
$t_{SU;DAT}$	Data Set-Up Time	0.25		0.1		µs
$t_{HD;DAT}$	Data Hold Time	0		0		µs
$t_{SU;STO}$	STOP Condition Set-Up Time	4.0		0.6		µs

41.7 Fan Tachometer Timing

FIGURE 41-15: FAN TACHOMETER INPUT TIMING

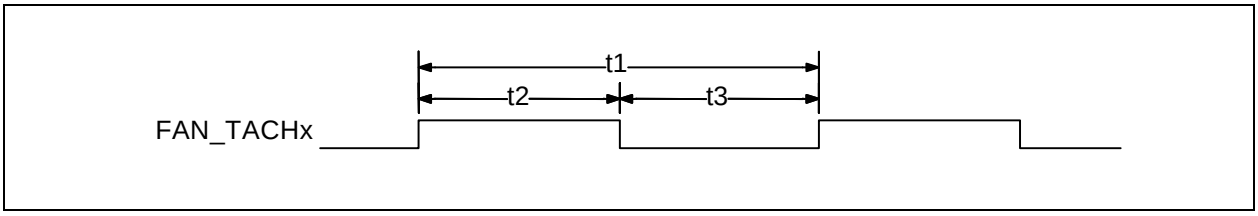


TABLE 41-10: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			µsec
t2	Pulse High Time				
t3	Pulse Low Time	10			

Note 41-1 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.

41.8 PS/2 Timing

FIGURE 41-16: PS/2 TRANSMIT TIMING

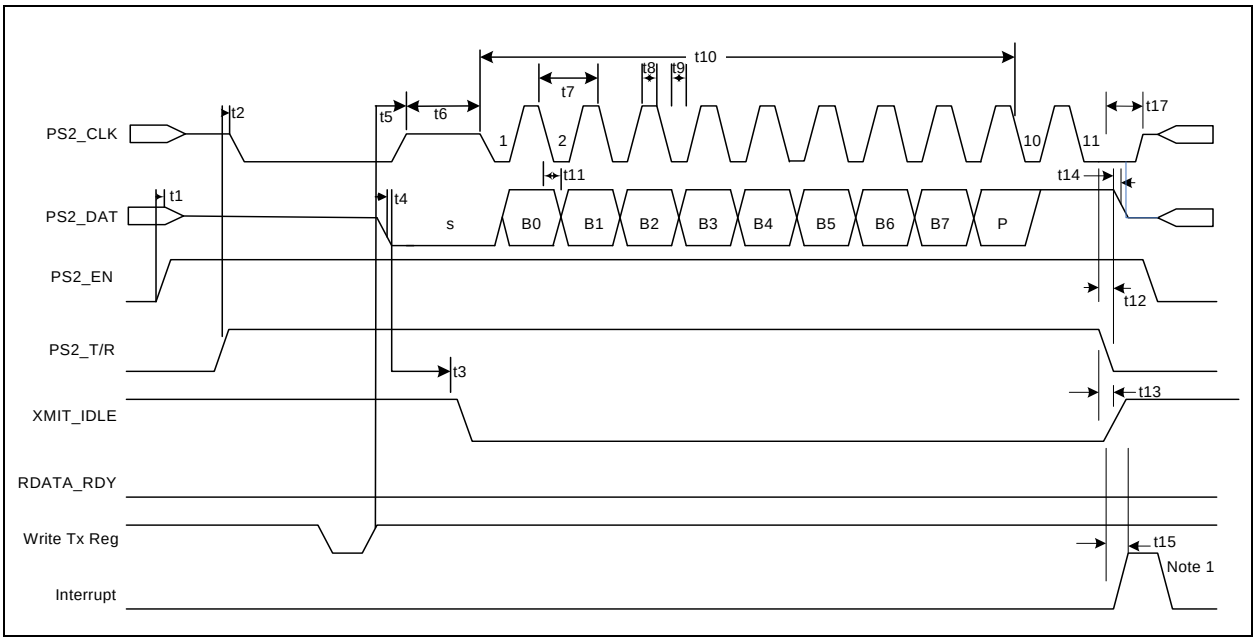


TABLE 41-11: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	PS2_T/R bit set to CLK driven low preparing the PS/2 Channel for data transmission.				
t3	CLK line floated to XMIT_IDLE bit de-asserted.			1.7	
t4	Trailing edge of WR to Transmit Register to DATA line driven low.	45		90	
t5	Trailing edge of EC WR of Transmit Register to CLK line floated.	90		130	ns
t6	Initiation of Start of Transmit cycle by the PS/2 channel controller to the auxiliary peripheral's responding by latching the Start bit and driving the CLK line low.	0.002		25.003	ms
t7	Period of CLK	60		302	µs
t8	Duration of CLK high (active)	30		151	
t9	Duration of CLK low (inactive)				
t10	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t11	DATA output by MEC1609/MEC1609i following the falling edge of CLK. The auxiliary peripheral device samples DATA following the rising edge of CLK.			1.0	µs
t12	Rising edge following the 11th falling clock edge to PS_T/R bit driven low.	3.5		7.1	µs
t13	Trailing edge of PS_T/R to XMIT_IDLE bit asserted.			500	ns
t14	DATA released to high-Z following the PS2_T/R bit going low.				
t15	XMIT_IDLE bit driven high to interrupt generated. Note1- Interrupt is cleared by writing a 1 to the status bit in Table 16-53, "GIRQ19 Source Register," on page 281.				
t17	Trailing edge of CLK is held low prior to going high-Z				

FIGURE 41-17: PS/2 RECEIVE TIMING

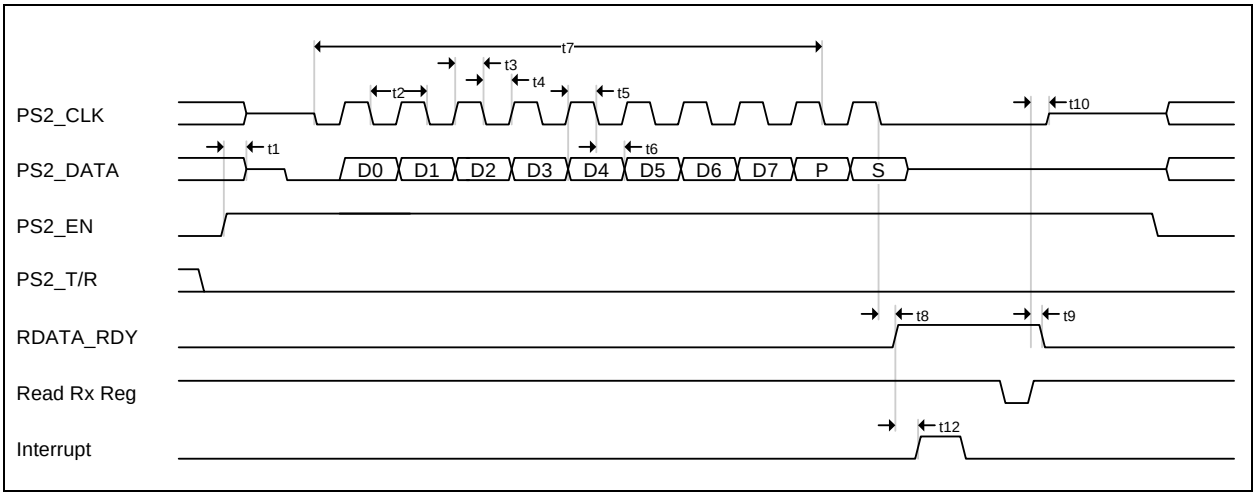


TABLE 41-12: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	Period of CLK	60		302	µs
t3	Duration of CLK high (active)	30		151	
t4	Duration of CLK low (inactive)				
t5	DATA setup time to falling edge of CLK. MEC1609/MEC1609i samples the data line on the falling CLK edge.	1			
t6	DATA hold time from falling edge of CLK. MEC1609/MEC1609i samples the data line on the falling CLK edge.	2			
t7	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t8	Falling edge of 11th CLK to RDATA_RDY asserted.			1.6	µs
t9	Trailing edge of the EC's RD signal of the Receive Register to RDATA_RDY bit de-asserted.			500	ns
t10	Trailing edge of the EC's RD signal of the Receive Register to the CLK line released to high-Z.				
t12	RDATA_RDY asserted an interrupt is generated.				

41.9 BC-Link Master Timing

FIGURE 41-18: BC-LINK TIMING

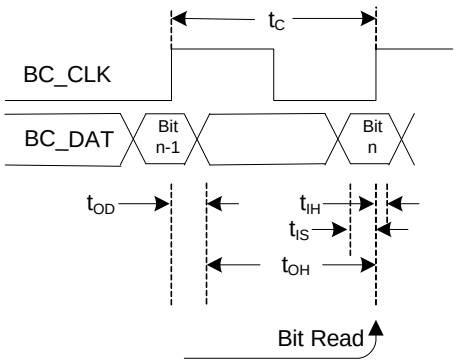


TABLE 41-13: BC-LINK MASTER TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_c(High Speed)	High Spec BC Clock Frequency		21.5	22.6	MHz
	High Spec BC Clock Period	44.3	46.7		ns
t_c(Low Speed)	High Spec BC Clock Frequency		2.93	3.08	Mhz
	High Spec BC Clock Period	324.7	341.0		ns
t_{OD}	BC-Link Master DATA output delay after rising edge of CLK.			10	nsec
t_{OH}	Data hold time after falling edge of CLK	1 CLK - $t_{OD\text{-}max}$			nsec
t_{IS}	BC-Link Master DATA input setup time before rising edge of CLK.	15			nsec
t_{IH}	BC-Link Master DATA input hold time after rising edge of CLK.	0			nsec

Note 41-2 The (t_{IH} in Table 41-13) BC-Link Master DATA input must be stable before next rising edge of CLK.

Note 41-3 The BC-Link Clock frequency is limited by the application usage model (see Note 37-1 on page 466 & Table 37-2 on page 467.) The BC-Link Clock frequency is controlled by the BC-Link Clock Select Register on page 470. The tc(High Speed) parameter implies both BC-link master and companion devices are located on the same circuit board and the BC-Link Clock Select Register set to 02h. The tc(Low Speed) parameter implies the BC-link master and companion devices are located on separate circuit boards connected by 12 inch ribbon cable and the BC-Link Clock Select Register set to 21h.

APPLICATION NOTE: The timing budget equation is as follows for data from BC-Link slave to master:

Tc > TOD(master-clk) + Tprop(clk) +TOD(slave) + Tprop(slave data) + TIS(master).

41.10 Serial Peripheral Interface (SPI) Timings

41.10.1 SPI CLOCK TIMING

FIGURE 41-19: SPI CLOCK TIMING

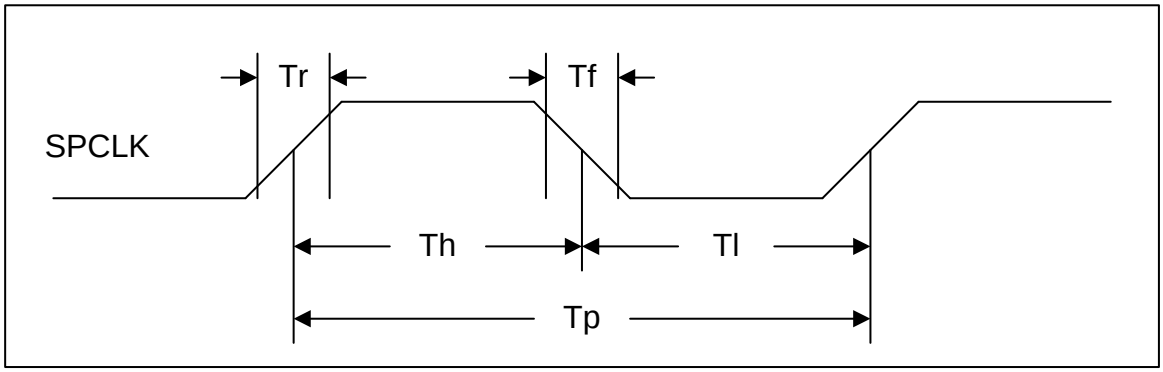


TABLE 41-14: SPI CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
Tr	SPI Clock Rise Time. Measured from 10% to 90%.			10% of SPCLK Period	ns
Tf	SPI Clock Fall Time. Measured from 90% to 10%.			10% of SPCLK Period	ns
Th/Tl	SPI Clock High Time/SPI Clock Low Time	40% of SPCLK Period	50% of SPCLK Period (Note 41-4)	60% of SPCLK Period	ns
Tp	SPI Clock Period – As selected by.SPI Clock Generator Register (SPICG) on page 430	15.50		62492.25	ns

Note 41-4 IIn the MEC1609/MEC1609i, the General Purpose Serial Peripheral Interface (GP-SPI) pins are 8 mA buffers. The maximum SPCLK pin clock frequency is 16.128MHz for all modes. Limited functionality is available at 32.26 MHz and although the block can be programmed for higher frequencies performance may not be maintained. See TABLE 31-14: on page 430 and Section 31.9.5.5, "Limits of SPI configurations," on page 424.

41.10.2 SPI SETUP AND HOLD TIMES

FIGURE 41-20: SPI SETUP AND HOLD TIMES

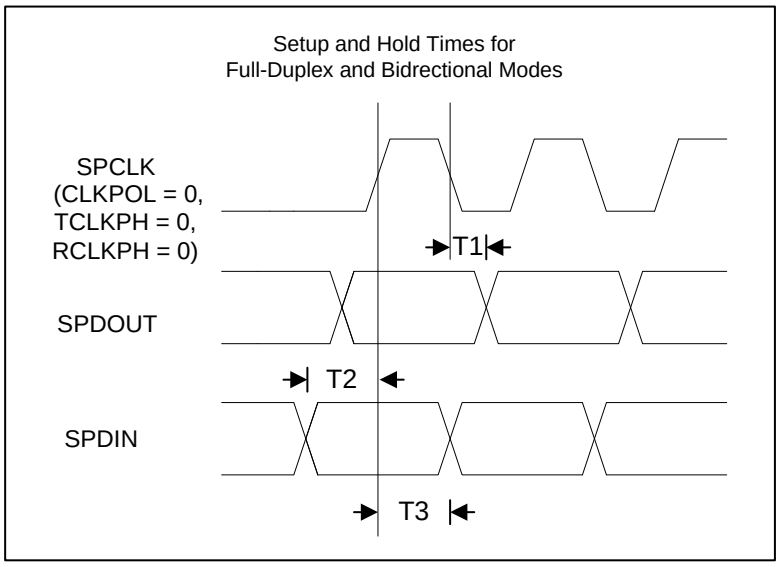


TABLE 41-15: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			20	ns
T2	Data IN Setup Time	20			ns
T3	Data IN Hold Time	0			ns

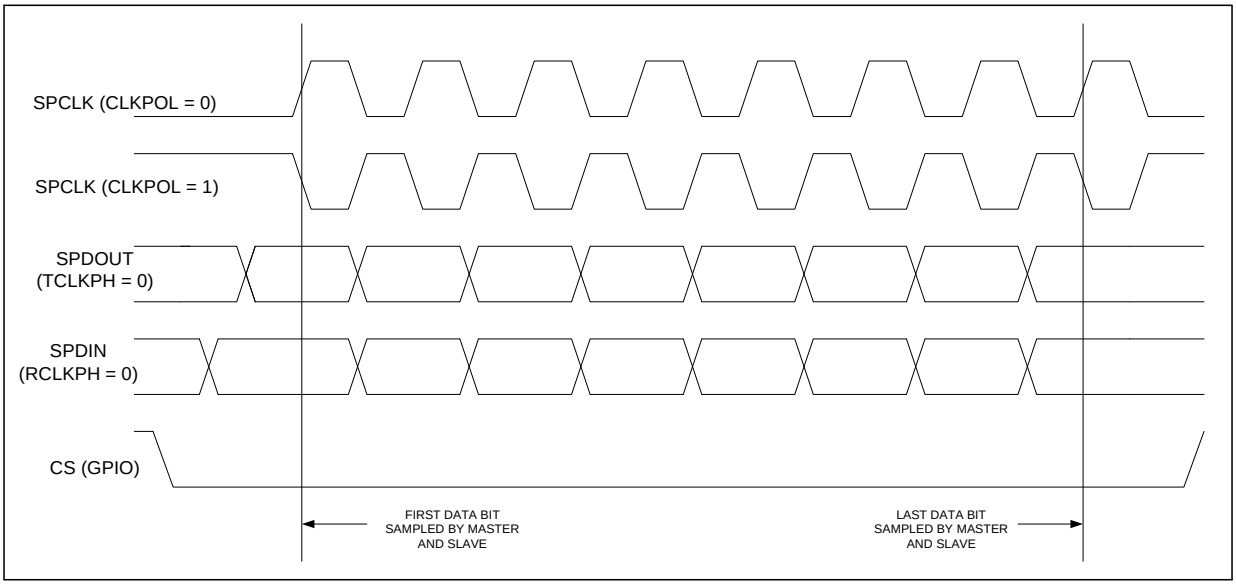
41.10.3 SPI INTERFACE TIMINGS

The following timing diagrams represent a single-byte transfer over the SPI interface using different SPCLK phase settings. Data bits are transmitted in bit order starting with the MSB (LSBF='0') or the LSB (LSBF='1'). See the SPICR - SPI Control Register on page 426 for information on the LSBF bit. The CS signal in each diagram is a generic bit-controlled chip select signal required by most peripheral devices. This signal and additional chip selects can be GPIO controlled. Note that these timings for Full Duplex Mode are also applicable to Half Duplex (or Bi-directional) mode.

41.10.3.1 SPI Interface Timing – Full Duplex Mode (TCLKPH = 0, RCLKPH = 0)

In this mode, data is available immediately when a device is selected and is sampled on the first and following odd SPCLK edges by the master and slave.

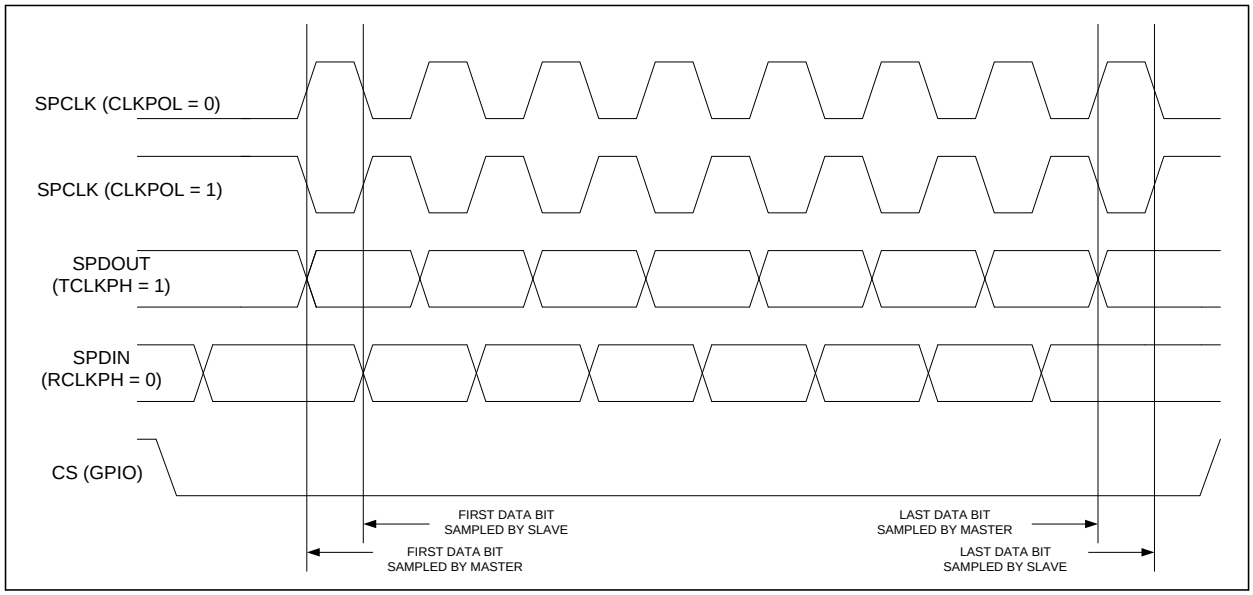
FIGURE 41-21: INTERFACE TIMING, FULL DUPLEX MODE (TCLKPH = 0, RCLKPH = 0)



41.10.3.2 SPI Interface Timing - Full Duplex Mode (TCLKPH = 1, RCLKPH = 0)

In this mode, the master requires an initial SPCLK edge before data is available. The data from slave is available immediately when the slave device is selected. The.data is sampled on the first and following odd edges by the master. The data is sampled on the second and following even SPCLK edges by the slave.

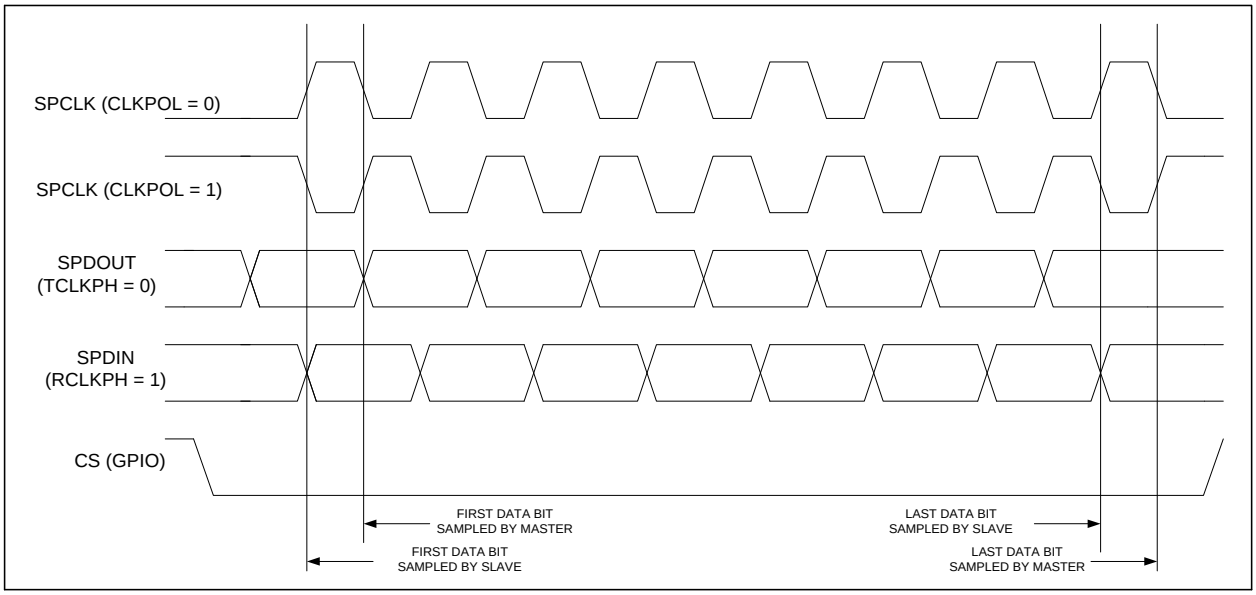
FIGURE 41-22: SPI INTERFACE TIMING, FULL DUPLEX MODE (TCLKPH = 1, RCLKPH = 0)



41.10.3.3 SPI Interface Timing - Full Duplex Mode (TCLKPH = 0, RCLKPH = 1)

In this mode, the data from slave is available immediately when the slave device is selected. The slave device requires an initial SPCLK edge before data is available. The data is sampled on the second and following even SPCLK edges by the master. The data is sampled on the first and following odd edges by the slave.

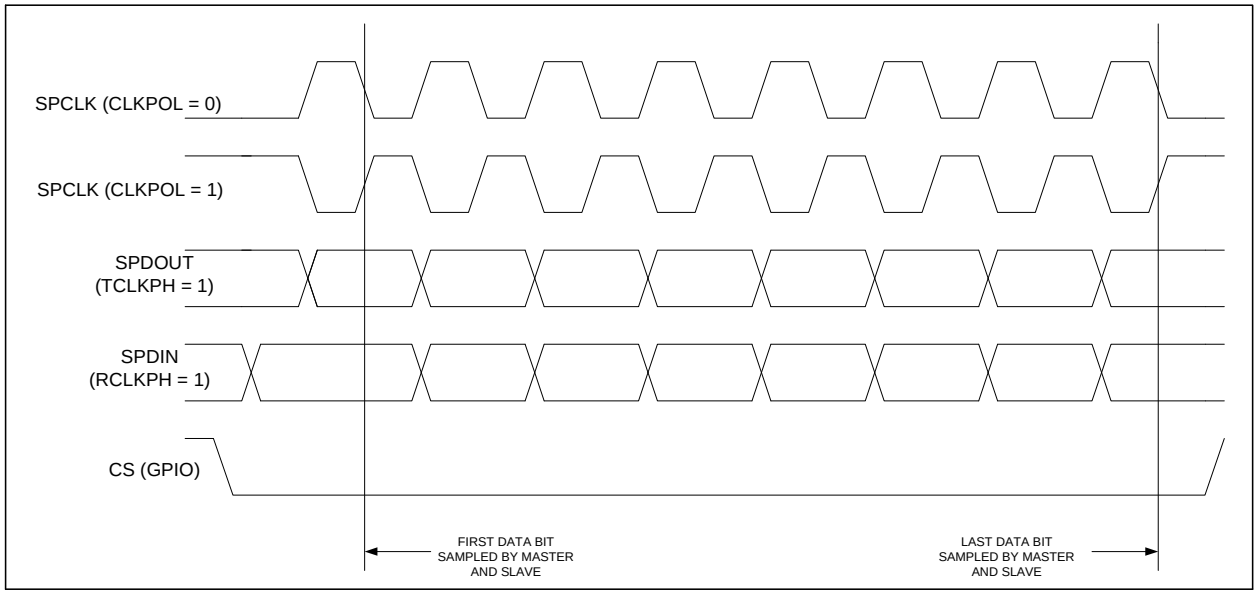
FIGURE 41-23: SPI INTERFACE TIMING, FULL DUPLEX MODE (TCLKPH = 0, RCLKPH = 1)



41.10.3.4 SPI Interface Timing - Full Duplex Mode (TCLKPH = 1, RCLKPH = 1)

In this mode, the master and slave require an initial SPCLK edge before data is available. Data is sampled on the second and following even SPCLK edges by the master and slave.

FIGURE 41-24: SPI INTERFACE TIMING - FULL DUPLEX MODE (TCLKPH = 1, RCLKPH = 1)



41.11 JTAG Interface Timing

FIGURE 41-25: JTAG POWER-UP & ASYNCHRONOUS RESET TIMING

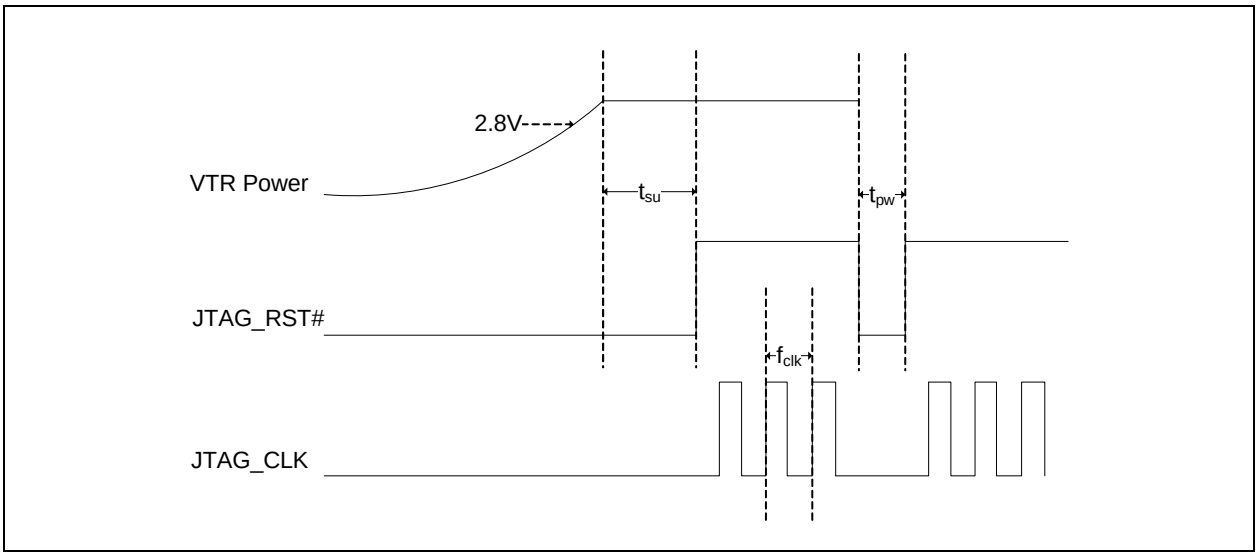


FIGURE 41-26: JTAG SETUP & HOLD PARAMETERS

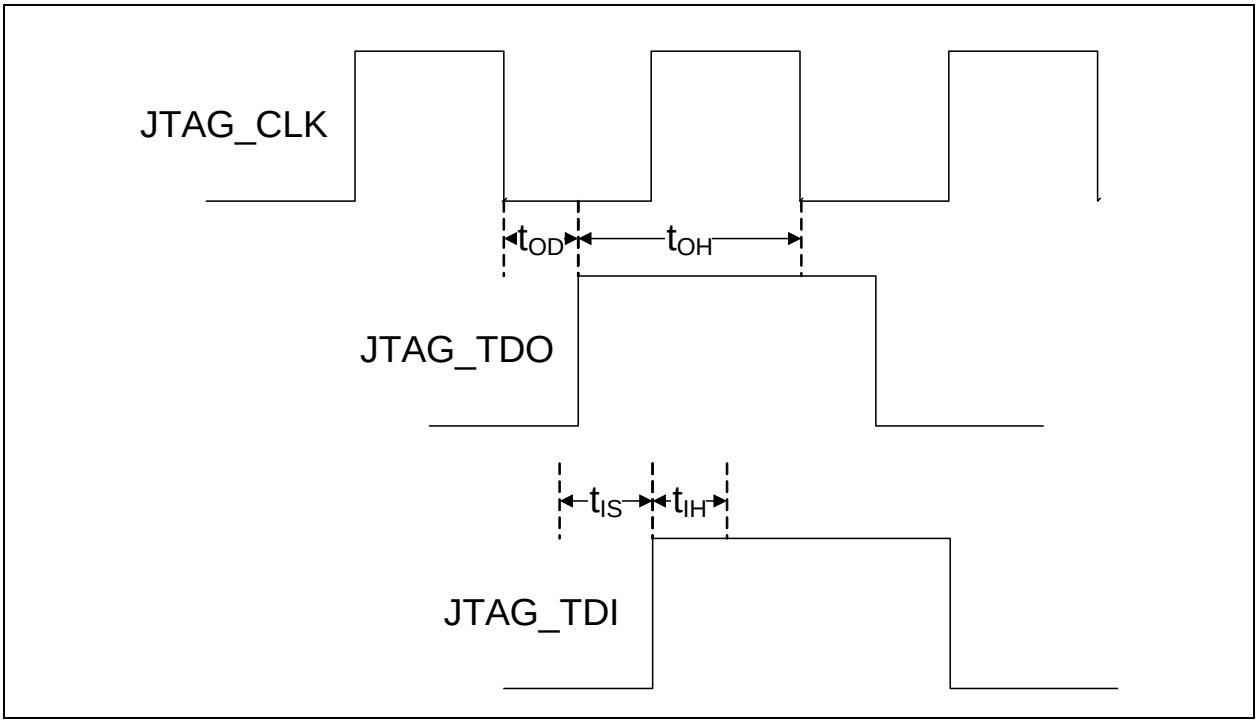


TABLE 41-16: JTAG INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{su}	JTAG_RST# de-assertion after VTR power is applied	500			µs
t_{pw}	JTAG_RST# assertion pulse width	500			nsec
f_{clk}	JTAG_CLK frequency (see note)			8	MHz
t_{OD}	TDO output delay after falling edge of TCLK.	5		18	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.	7			nsec

Note 41-1 f_{clk} is the maximum frequency to access a JTAG Register. Additional JTAG_CLK frequency constraints are described in JTAG and XNOR, Section 39.7.2, "Clocks," on page 484.

41.12 Serial Debug Port Timing

FIGURE 41-27: SERIAL DEBUG PORT TIMING PARAMETERS

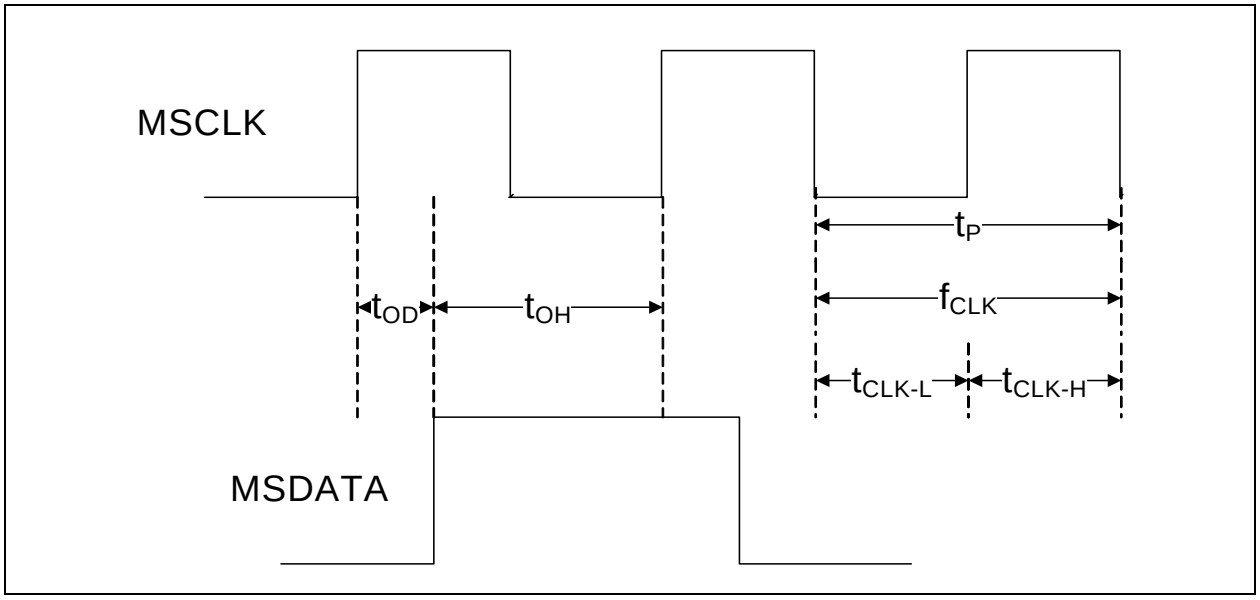


TABLE 41-17: SERIAL DEBUG PORT INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
f_{clk}	MSCLK frequency (see note)		32	48	MHz
t_{OD}	MSDATA output delay after falling edge of MSCLK.			5	nsec
t_{OH}	MSDATA hold time after falling edge of TCLK	1 MSCLK - t_{OD}			nsec
t_P	MSCLK Period.	1/fclk			nsec
t_{CLK-L}	MSCLK Low Time	15		fclk/2	nsec
t_{CLK-H}	MSCLK high Time (see Note 41-1)	fclk/2 - 15		fclk/2	nsec

Note 41-1 When the EC_CLK_DIV is an odd number value greater than 2h, then $t_{CLK-L} = t_{CLK-H} + 15ns$. When the EC_CLK_DIV is 0h, 1h, or an even number value greater than 2h, then $t_{CLK-L} = t_{CLK-H}$.

42.0 REFERENCE DOCUMENTS

This document was created using the following parent documents:

1. Intel Low Pin Count Specification, Revision 1.0, September 29, 1997
2. PCI Local Bus Specification, Revision 2.2, December 18, 1998
3. Advanced Configuration and Power Interface Specification, Revision 1.0b, February 2, 1999
4. System Management Bus Specification, Revision 1.1, December 11, 1998.
5. Plug and Play ISA Specification, Version 1.0a, Intel Corp. and Microsoft Corp., May 5, 1994
6. I2C-BUS Specification, Version 2.1, January 2000.
7. SMBus Controller Core Interface, Revision 2.0, v2.17, Core-Level Architecture Specification, 2/11/09
8. ECE1077 MEC-04 Keyboard Scan Extension, Product Architecture Specification, Rev 0.23, January 12, 2006, Confidential
9. Intel® 82801DBM I/O Controller Hub 4 Mobile (ICH4-M), Datasheet, Order Number: 252337-001, Intel Corp., January 2003
10. BC-Link Specification, Revision1.02, dated September 05, 2007
11. IEEE Std 1149.1
12. AN 19.3, PCB Layout Guide for MEC1609, Revision 0.4 (01-21-09), Confidential Application Note
13. PECI Interface Core, Rev. 1.1, Core-Level Architecture Specification, SMSC Confidential
14. PCI Mobile Design Guide, Version 1.1, PCI-SIG, December 18, 1998.

APPENDIX A: DATA SHEET REVISION HISTORY

Revision	Section/Figure/Entry	Correction
DS00002485A (06-21-17)	Document Release	

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X]	-	XXX	-	[X](1)
Device	Temperature Range		Package		Tape and Reel Option

Device:	MEC1609, MEC1609i	
Temperature Range:	Blank = 0°C to +70°C i = -40°C to +85°C	(Extended Commercial) (Industrial)
Package:	PZV = 144-pin TFBGA PZP = 144-pin LFBGA	
Tape and Reel Option:	Blank = Standard packaging (tray) TR = Tape and Reel(1)	

Examples:

c) MEC1609-PZV
144-pin TFBGA (7mm x 7mm, 0.5 pitch)
RoHS Compliant package

d) MEC1609-PZP
144-pin TFBGA (10mm x 10mm, 0.8 pitch)
RoHS Compliant package

e) MEC1609i-PZV
Industrial temperature,
144-pin TFBGA (7mm x 7mm, 0.5 pitch)
RoHS Compliant package

f) MEC1609i-PZP
Industrial temperature,
144-pin TFBGA (10mm x 10mm, 0.8 pitch)
RoHS Compliant package

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option. Reel size is 4,000.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 9781522418382

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820