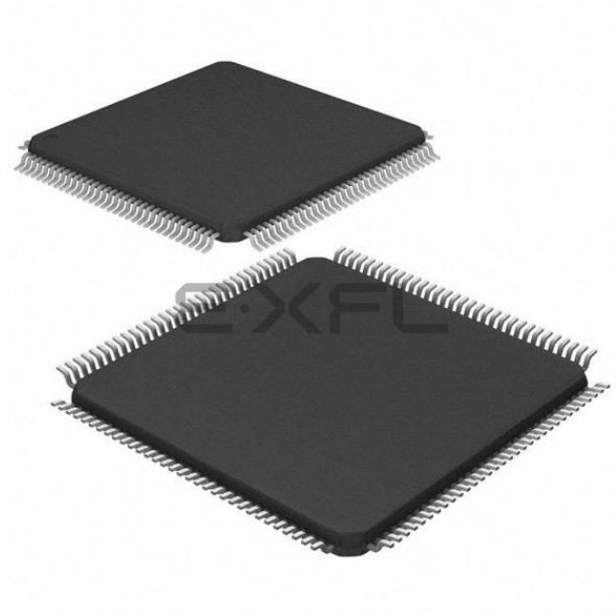

Welcome to E-XFL.COM

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

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

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Active
Applications	Keyboard and Embedded Controller
Core Processor	MIPS32® M14K™
Program Memory Type	External Program Memory
Controller Series	-
RAM Size	160KB
Interface	I²C, LPC, SMBus, SPI, UART
Number of I/O	106
Voltage - Supply	1.71V ~ 3.465V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	128-TQFP
Supplier Device Package	128-VTQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/mec1416-nu

1.0 GENERAL DESCRIPTION

The MEC140x/1x is a family of keyboard and embedded controller designs customized for notebooks and tablet platforms. The MEC140x/1x family is a highly-configurable, mixed signal, advanced I/O controller architecture. Every device in the family incorporates a 32-bit MIPS32 M14K Microcontroller core with a closely-coupled SRAM for code and data. A secure boot-loader is used to download the custom firmware image from the system's shared SPI Flash device, thereby allowing system designers to customize the device's behavior.

The MEC140x/1x products may be configured to communicate with the system host through one of three host interfaces: Intel Low Pin Count (LPC), eSPI, or I2C. Note that this functionality is product dependent. To see which features apply to a specific part in the family see Products on page 3. The document defines the features for all devices in the family.

The MEC140x/1x products are designed to operate as either a stand-alone I/O device or as an EC Base Component of a split-architecture Advanced I/O Controller system which uses BC-Link communication protocol to access up to two BC bus companion components. The BC-Link protocol is peer-to-peer providing communication between the MEC140x/1x embedded controller and registers located in a companion device.

The MEC140x/1x is directly powered by a minimum of two separate suspend supply planes (VBAT and VTR) and senses a third runtime power plane (VCC) to provide "instant on' and system power management functions. In addition, this family of products has the option to connect the VTR_33_18 power pin to either a 3.3V VTR power supply or a 1.8V power supply. This option may only be used with the eSPI Host Interface or the I2C Host Interface. In systems using the I2C Host Interface, ten GPIOs are powered by VTR_33_18, thereby allowing them to operate at either 3.3V or 1.8V. All the devices are equipped with a Power Management Interface that supports low-power states and are capable of operating in a Connected Standby system.

The MEC140x/1x family of devices offer a software development system interface that includes a Trace FIFO Debug port, a host accessible serial debug port with a 16C550A register interface, a Port 80 BIOS Debug Port, and an In-circuit Serial Programming (ICSP) interface.

1.1 Boot ROM

Following the release of the EC_PROC_RESET# signal, the processor will start executing code in the Boot ROM. The Boot ROM executes the SPI Flash Loader, which downloads User Code from an external SPI Flash and stores it in the internal Code RAM. Upon completion, the Boot ROM jumps into the User Code and starts executing.

1.2 Initialize Host Interface

By default, this device powers up all the interfaces, except the VBAT powered interfaces and select signals, to GPIO inputs. The Boot ROM is used to download code from an external flash via either the Shared Flash Interface, the eSPI flash channel or the Private Flash Interface. The downloaded code must configure the device's pins according to the platform's needs. This includes initializing the Host Interface.

Once the device is configured for operation, the downloaded code must deassert the system's RSMRST# (Resume Reset) signal. Any GPIO may be selected for the RSMRST# function. This is up to the system board designer. The only requirement is that the board designer attach an external pull-down on the GPIO pin being used for the RSMRST# function. This will ensure the RSMRST# pin is asserted low by default and does not glitch during power-up.

This family of devices has up to three Host Interface options. It may be configured as an LPC Device, an eSPI Device, or I2C device. See Products on page 3 for the features supported in each device.

On a VTR POR, all the host interface pins default to GPIO inputs.

1.2.1 CONFIGURE LPC INTERFACE

The downloaded firmware must configure the GPIO Pin Control registers for the LPC alternate function, configure the LPC Base Address Register (BAR), and activate the LPC block.

Example:

- GPIO034 Pin Control Register = 0x1000; //ALT FUNC1 – PCI_CLK
- GPIO040 Pin Control Register = 0x1000; //ALT FUNC1 – LAD0
- GPIO041 Pin Control Register = 0x1000; //ALT FUNC1 – LAD1
- GPIO042 Pin Control Register = 0x1000; //ALT FUNC1 – LAD2
- GPIO043 Pin Control Register = 0x1000; //ALT FUNC1 – LAD3
- GPIO044 Pin Control Register = 0x1000; //ALT FUNC1 – LFRAME_N

MEC140x								
VTQFP Pin#	**Mux**	**Signal Name**	**Buffer Type**	**Default Buffer Operation**	**Signal Power Well**	**Emulated Power Well**	**Gated State**	**Notes**
2	3	Reserved	Reserved		Reserved	Reserved		
2	Strap							
3	Default: 0	GPIO001	PIO	I-4	VTR	VTR/VCC	No Gate	
3	1	SPI_CS#	PIO		VTR	VTR	Reserved	
3	2	32KHZ_OUT	PIO		VTR	VTR	Reserved	
3	3	Reserved	Reserved		Reserved	Reserved		
3	Strap							
4	Default: 0	GPIO002	PIO	I-4	VTR	VTR/VCC	No Gate	
4	1	PWM7	PIO		VTR	VTR	Reserved	
4	2	Reserved	Reserved		Reserved	Reserved		
4	3	Reserved	Reserved		Reserved	Reserved		
4	Strap							
5		VTR	PWR		PWR	PWR		
5								
5								
5								
5	Strap							
6	Default: 0	GPIO005	PIO	I-4	VTR	VTR/VCC	No Gate	
6	1	SMB00_DATA	PIO		VTR	VTR	High	Note 4
6	2	SMB00_DATA18	PIO		VTR	VTR	High	Note 11
6	3	KSI2	PIO		VTR	VTR	Low	Note 15
6	Strap							
7	Default: 0	GPIO006	PIO	I-4	VTR	VTR/VCC	No Gate	
7	1	SMB00_CLK	PIO		VTR	VTR	High	Note 4
7	2	SMB00_CLK18	PIO		VTR	VTR	High	Note 11
7	3	KSI3	PIO		VTR	VTR	Low	Note 15
7	Strap							
8	Default: 0	GPIO007	PIO	I-4	VTR	VTR/VCC	No Gate	
8	1	SMB01_DATA	PIO		VTR	VTR	High	Note 4
8	2	SMB01_DATA18	PIO		VTR	VTR	High	Note 11
8	3	Reserved	Reserved		Reserved	Reserved		
8	Strap							
9	Default: 0	GPIO010	PIO	I-4	VTR	VTR/VCC	No Gate	
9	1	SMB01_CLK	PIO		VTR	VTR	High	Note 4
9	2	SMB01_CLK18	PIO		VTR	VTR	High	Note 11
9	3	Reserved	Reserved		Reserved	Reserved		
9	Strap							
10	Default: 0	GPIO011	PIO	I-4	VTR	VTR/VCC	No Gate	
10	1	nSMI	PIO		VTR	VTR	Reserved	
10	2	nEMI_INT	PIO		VTR	VTR	Reserved	
10	3	Reserved	Reserved		Reserved	Reserved		
10	Strap							

3.4.3 POWER GOOD SIGNALS

The power good timing and thresholds are defined in the Section 43.1, "Voltage Thresholds and Power Good Timing," on page 501.

TABLE 3-3: POWER GOOD SIGNAL DEFINITIONS

Power Good Signal	Description	Source
VTRGD	VTRGD is an internal power good signal used to indicate when the VTR rail is on and stable.	VTRGD is asserted following a delay after the VTR power well exceeds its preset voltage threshold. VTRGD is de-asserted as soon as either of these voltages drop below this threshold. **Note:** See Section 43.1.1, "VTR Threshold and VTRGD Timing," on page 501.
VCC_PWRGD	VCC_PWRGD is used to indicate when the main power rail voltage is on and stable.	VCC_PWRGD Input pin

3.4.4 SYSTEM POWER SEQUENCING

The following table defines the behavior of the Power Sources in each of the defined ACPI power states.

TABLE 3-4: TYPICAL POWER SUPPLIES VS. ACPI POWER STATES

Supply Name	ACPI Power State						Description
	S0 (FULL ON)	S1 (POS)	S3 (STR)	S4 (STD)	S5 (Soft Off)	G3 (MECH Off)	
VTR_33_18	ON	ON	ON/OFF	ON/OFF	ON/OFF	OFF	LPC/eSPI Host Interface Power Supply.
VTR	ON	ON	ON	ON	ON	OFF	“Always-on” Supply. (Note 3-4)
VBAT	ON	ON	ON	ON (Note 3-5)	ON (Note 3-5)	ON (Note 3-5)	Battery Back-up Supply

Note 3-4 VTR power supply is always on while the battery pack or ac power is applied to the system.

Note 3-5 This device requires that the VBAT power is on when the VTR power supply is on. External circuitry, a diode isolation circuit, is implemented on the motherboard to extend the battery life. This external circuitry ensures the VBAT pin will derive power from the VTR power well when it is on. Therefore, the VBAT supply will never appear to be off when the VTR rail is on. See APPLICATION NOTE: on page 65.

3.5 Clocks

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

3.5.1 RAW CLOCK SOURCES

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

TABLE 4-13: CONFIGURATION REGISTER SUMMARY

Register Name	Offset	Size	Notes
LPC Activate Register	30h	8	
SIRQ Configuration Register Format	40h - 4Fh	8	
I/O Base Address Registers (IO_BARs)	60h - 9Fh See TABLE 4-15:	32	
SRAM Memory BAR	A0h	32	
SRAM Memory BAR Configuration	A4h	32	
Device Memory Base Address Registers (DEV_MEM_BARs)	C0h - FFh See TABLE 4-16:	48	

4.9.1 LPC ACTIVATE REGISTER

The LPC 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 device. The Embedded Controller can set or clear the Activate bit at any time.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	RESERVED	RES	-	-
0	ACTIVATE 1= Activate When this bit is 1, the LPC Logical Device is powered and functional. 0= Deactivate When this bit is 0, the logical device is powered down and inactive. Except for the LPC 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 4.11.4, "EC Clock Control Register," on page 123). LPC bus output pads will be tri-stated.	R/W	0b	nSYSRST

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

4.9.2 SERIAL IRQ CONFIGURATION REGISTERS

The LPC Controller implements 16 IRQ channels that may be configured to be asserted by any logical device.

- For a description of the SIRQ Configuration Register format see Table 4-14, "SIRQ Interrupt Configuration Register Map," on page 114.
- For a summary of the SIRQ IRQ Configuration registers implemented see Table 4-15, "I/O Base Address Registers," on page 117.
- For a list of the SIRQ sources see Table 4-11, "Logical Device SIRQ Routing," on page 110.

4.10.2 DATA REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA The DATA register, which is part of the Configuration Port, is used to read or write data to the register currently being selected by the INDEX Register. **Note:** For a description of accessing the Configuration Port see Section 4.8.3, "Configuration Port," on page 105	R/W	0h	nSYSRST

4.11 EC-Only Registers

Note: EC-Only registers are not accessible by the LPC interface.

The registers listed in Table 4-20, "EC-Only Register Summary" are for a single instance of the LPC Interface. Their addresses are defined as a relative offset to the host base address defined in TABLE 4-19:.

The following table defines the fixed host base address for each LPC Interface instance.

TABLE 4-19: EC-ONLY REGISTER ADDRESS RANGE TABLE

INSTANCE NAME	INSTANCE NUMBER	HOST	ADDRESS SPACE	BEGIN ADDRESS
LPC Interface	0	EC	32-bit internal address space	000F_3100h

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

TABLE 4-20: EC-ONLY REGISTER SUMMARY

Offset	Register Name
04h	LPC Bus Monitor Register
08h	Host Bus Error Register
0Ch	EC SERIRQ Register
10h	EC Clock Control Register
14h	Test Register
18h	Test Register
20h	I/O BAR Inhibit Register
24h	Reserved
28h	Reserved
2Ch	Reserved
30h	LPC BAR Init Register
40h	Device Memory BAR Inhibit Register
FCh	SRAM Memory Host Configuration Register

Note 4-13 Some Test registers are read/write registers. Modifying these registers may have unwanted results.

STEPS TO SET UP A PARTICULAR GIRQ GROUPING OF INTERRUPTS TO VECTOR TO AN ISR IN DISAGGREGATED/JT MODE.

1. Determine a location in code space to contain a mini-jump table, of size 31 entries or less, depending on how populated a particular GIRQ is (i.e. 15 populated sources = 15 jump table entries in SRAM).
2. Build up to 31 ISRs, one for each interrupt source in this GIRQ. The jump table gets populated with jump instructions the locations of these ISRs.
3. Program the 17-bit offset for the entry location of the mini-jump table into the GIRQ aggregator control/vector address register. EBASE must be programmed at 0xbfd0_0000.
4. (optional) Clear all source bits for the interrupts within GIRQ “n”.
5. Enable the individual interrupts within GIRQ “n” that you wish to be interrupt the processor.
6. Enable global interrupts in the processor.

ILLUSTRATIVE SCENARIO:

GIRQ #8 has 31 GPIOs from pins configured to generate interrupts that will be handled by an 31 ISRs labeled “GIRQ08_GPIO001_handler”, “GIRQ08_GPIO002_handler”, etc.

The 31 GPIOs are named (from GIRQ #8’s bit 0 through bit 30): GPIO001, GPIO002,….,GPIO030.

EBASE is at 0xbfd0_0000. Firmware places the jump table at address 0xbfd0_0500. The jump table gets populated with jump instructions to the 31 ISRs.

The firmware programs GIRQ #8’s aggregator control to 0x0000_0501 (bits 17:1 are the vector address, bit 0 is the GIRQ control to aggregate/dis-aggregate).

Firmware then sets each interrupt source priority to, say, 0x0 (2 bits of priority), which corresponds to priority level 1 to the processor. Then enables all interrupt lines by writing 0xffff_ffff to GIRQ #8’s interrupt “enable set” register address.

If GPIO029 later fires an interrupt to the controller, the controller will send an EIC vector of 0x5e8 with a requested interrupt priority level 1 to the processor. This causes the processor to vector to the 30th entry in the mini-jump table, which then jumps to the “GIRQ08_GPIO029_handler” code.

This address: 0x5e8 = vector base + 29*(vector spacing) which is by default 8 bytes.

Later, GPIO002 fires an interrupt to the controller, which causes the controller to send an EIC vector of 0x510 with a requested interrupt priority level 1 to the processor. This causes the processor to vector to the 3rd entry in the mini-jump table, which then jumps to the “GIRQ08_GPIO002_handler” code.

10.11.6 HYBRID MODE

The Hybrid is a combination of the aggregated and disaggregated modes.

Each GIRQ group has the option of operating in either aggregated mode or disaggregated mode. This mode is similar to the disaggregated mode, except the grouped GIRQs will OR their result through bit 0 of that GIRQ. Each GIRQx[n] Result Bit is assigned the priority-level that is programmed in the corresponding GIRQx[n] Priority bit. The Priority Encoder and Decision Logic generates the Vector for the active Result bit with the highest priority. If two or more Result bits are active with the same priority-level the lowest Result Bit wins.

The following diagram illustrates this selection process.

12.8.4 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 memory in the 32-bit internal address space can be used to implement a set of virtual registers. The Host is given direct read-only access to this address space, referred to as peek mode. The EC may read or write this memory as needed.
- Program downloading. Because the Instruction Closely Coupled Memory is implemented in the same 32-bit internal address space, the Embedded Memory Interface can be used by the Host to download new program segments for the EC in the upper 32KB SRAM. 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 32-bit internal address space, 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 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:	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.

12.9 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the **EMI**. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

TABLE 12-2: RUNTIME REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
EMI	0	EC	32-bit internal address space	000F_0000h
		LPC	I/O	Programmed BAR

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

12.9.13 APPLICATION ID REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
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.	R/W	0h	nSYSRST

12.10 EC-Only Registers

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

TABLE 12-4: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
EMI	0	EC	32-bit internal address space	000F_0100h

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

TABLE 12-5: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	HOST-to-EC Mailbox Register
01h	EC-to-HOST Mailbox Register
04h	Memory Base Address 0 Register
08h	Memory Read Limit 0 Register
0Ah	Memory Write Limit 0 Register
0Ch	Memory Base Address 1 Register
10h	Memory Read Limit 1 Register
12h	Memory Write Limit 1 Register
14h	Interrupt Set Register
16h	Host Clear Enable Register

14.0 ACPI EMBEDDED CONTROLLER INTERFACE (ACPI-ECI)

14.1 Introduction

The ACPI Embedded Controller Interface (ACPI-ECI) is a Host/EC Message Interface. The ACPI specification 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 (ACPI-ECI) provides a four byte full duplex data interface which is a superset of the standard ACPI Embedded Controller Interface (ACPI-ECI) one byte data interface. The ACPI Embedded Controller Interface (ACPI-ECI) defaults to the standard one byte interface.

The MEC140x/1x has two instances of the ACPI Embedded Controller Interface.

1. The EC host in TABLE 14-4: and TABLE 14-6: corresponds to the EC in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_EC.
2. The LPC host in TABLE 14-4: and TABLE 14-6: corresponds to the “System Host Interface to OS” in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_OS.

14.2 References

- Advanced Configuration and Power Interface Specification, Revision 4.0 June 16, 2009, Hewlett-Packard Corporation Intel Corporation Microsoft Corporation Phoenix Technologies Ltd. Toshiba Corporation

14.3 Terminology

TABLE 14-1: TERMINOLOGY

Term	Definition
ACPI_EC	The EC host corresponding to the ACPI specification interface to the EC.
ACPI_OS	The LPC host corresponding to the ACPI specification interface to the “System Host Interface to OS”. ACPI_OS terminology is not meant to distinguish the ACPI System Management from Operating System but merely the hardware path upstream towards the CPU.

14.4 Interface

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

14.12.5 ACPI OS COMMAND REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OSS_COMMAND Writes to the this register are aliased in the OS2EC Data EC Byte 0 Register. Writes to the this register also set the CMD and IBF bits in the OS STATUS OS Register	W	0h	nSYSRST

14.12.6 OS STATUS OS REGISTER

This read-only register is aliased to the EC STATUS Register on page 237. the EC STATUS Register on page 237 has read write access.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UD0B User Defined	R	0b	nSYSRST
6	SMI_EVT This bit is set when an SMI event is pending; i.e., the ACPI_EC is requesting an SMI query; This bit is cleared when no SMI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the ACPI_EC has detected an internal event that requires system management interrupt handler attention. The ACPI_EC sets this bit before generating an SMI. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.	R	0b	nSYSRST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	WRITE_CMD This 8-bit register is write-only and is an alias of the register at offset 0h. When written, the C/D bit in the Keyboard Status Read Register is set to '1', signifying a command, and the IBF in the same register is set to '1'. When the Runtime Register at offset 4h is read by the Host, it functions as the Keyboard Status Read Register.	W	0h	nSYSRST

16.14.2 EC_HOST DATA / AUX DATA REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	READ_DATA This 8-bit register is read-only. When read by the Host, the PCOBF and/or AUXOBF interrupts are cleared and the OBF flag in the status register is cleared.	R	0h	nSYSRST

16.14.3 KEYBOARD STATUS READ REGISTER

This register is a read-only alias of the EC Keyboard Status Register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	UD2 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	nSYSRST
5	AUXOBF Auxiliary Output Buffer Full. This bit is set to "1" whenever the EC writes the EC AUX Data Register. This flag is reset to "0" whenever the EC writes the EC2Host Data Register.	R	0h	nSYSRST
4	UD1 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	nSYSRST

17.11.4 PROGRAMMABLE BAUD RATE GENERATOR MSB REGISTER

Offset	01h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	BAUD_CLK_SEL 0=If CLK_SRC is '0', the baud clock is derived from the 1.8432MHz_Clk. If CLK_SRC is '1', this bit has no effect 1=If CLK_SRC is '0', the baud clock is derived from the 24MHz_Clk. If CLK_SRC is '1', this bit has no effect	R/W	0h	RESET
6:0	BAUD_RATE_DIVISOR_MSB See Section 17.9.1, "Programmable Baud Rate".	R/W	0h	RESET

17.11.5 INTERRUPT ENABLE REGISTER

The lower four bits of this register control the enables of the five interrupt sources of the Serial Port interrupt. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of this register. Similarly, setting the appropriate bits of this register to a high, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and disables any Serial Port interrupt out of the MEC140x/1x. All other system functions operate in their normal manner, including the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are described below.

Offset	01h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	Reserved	R	-	-
3	EMSI This bit enables the MODEM Status Interrupt when set to logic "1". This is caused when one of the Modem Status Register bits changes state.	R/W	0h	RESET
2	ELSI This bit enables the Received Line Status Interrupt when set to logic "1". The error sources causing the interrupt are Overrun, Parity, Framing and Break. The Line Status Register must be read to determine the source.	R/W	0h	RESET
1	ETHREI This bit enables the Transmitter Holding Register Empty Interrupt when set to logic "1".	R/W	0h	RESET
0	ERDAI This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic "1".	R/W	0h	RESET

19.7 Low Power Modes

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

The timer operates off of the 32KHz_Clk, and therefore will operate normally when 48 MHz Ring Oscillator is stopped. The sleep enable input has no effect on the RTOS Timer and the clock required output is only asserted during register read/write cycles for as long as necessary to propagate updates to the block core.

19.7.1 SLEEP INTERFACE - SYSTEM CLOCK

The RTOS Timer is designed to always operate in its lowest functional power consumption state. In addition, it can be commanded to enter a lower power state via the Sleep Enable signal. The block notifies the chip's power management circuitry when it is in its low power state by driving the Clock Required signal low. The following table defines all the blocks Power States associated with the System Clock.

Note:	The logic clocked by the system clock is considered to be in the idle state when the host is not accessing the register interface.

TABLE 19-5: RTOS Timer - SYSTEM CLOCK POWER STATES

Power State	Block Enable Bit	Sleep Enable	Clock Required	Description
Idle	x	x	0	Block is idle and operating in its lowest power consumption state. The 48 MHz Ring Oscillator is not used in this state. The block automatically enters this state anytime it is not performing a function requiring this clock source (e.g., Register accesses).
Operating	x	x	1	Block is not idle. This block will assert Clock Required signal only during register access and when it needs to generate interrupt. The sleep_en signal has no effect on this clock requirement.

Note: The RTOS Timer Registers are readable and writable in all defined Power States.

19.7.2 WAKING FROM LOW POWER STATES

The chip Power, Clocks, and Resets logic is responsible for monitoring wake events that turn on 48 MHz Ring Oscillator. The RTOS_TIMER interrupt event is a wake-capable event that may be used to turn on 48 MHz Ring Oscillator.

19.8 Description

The RTOS Timer is a very basic timer with simple down counter functionality with auto-reload and halt features. The timer counts with Timer Clock when the timer is programmed with pre-load value.

The counter can be configured as one-shot timer by not setting the Auto Reload bit. The timer will load the value of the pre-load register and start to count down when the Timer Start bit is asserted by the firmware. The timer will generate interrupt when the counter transitions from count = 1 to count = 0 as defined in the Interrupt Generation section.

If the timer is needed again with same pre-load value, firmware has to only set the Timer Start bit. This will restart the timer again.

The counter can also be programmed as continuous running mode by enabling the Auto Reload bit. In this mode counter reloads itself every time timer equals 0. The timer also generates interrupt as defined in the interrupt section.

If the RTOS Timer Pre-Load register is written when the counter is counting, the new preload value will take effect only when the counter reaches 0 if the auto-reload bit has been set.

If the RTOS Timer Pre-Load register is programmed with 32'h0 while the Timer is counting, the Timer will continue to count until it counts to 0. Then the Timer Start bit will be cleared. If the Timer Start bit is written when the RTOS Timer Pre-Load register is 0, the Timer Start bit will be self-cleared.

MEC141x						
GPIO Name (Octal)	**Pin Control Register Offset (Hex)**	**Pin Control Register Default (Hex)**	**Default Function**	**Pin Control Register 2 Offset (Hex)**	**Pin Control Register 2 Default (Hex)**	**Default Drive Strength (mA)**
GPIO067	00DC	00000000	GPIO067	5DC	00000010	4
GPIO100	0100	00000000	GPIO100	5E0	00000010	4
GPIO101	0104	00000000	GPIO101	5E4	00000010	4
GPIO102	0108	00000000	GPIO102	5E8	00000010	4
GPIO103	010C	00000000	GPIO103	5EC	00000010	4
GPIO104	0110	00000000	GPIO104	5F0	00000010	4
GPIO105	0114	00000000	GPIO105	5F4	00000010	4
GPIO106	0118	00000000	GPIO106	5F8	00000010	4
GPIO107	011C	00000000	GPIO107	5FC	00000010	4
GPIO110	0120	00000000	GPIO110	600	00000010	4
GPIO111	0124	00000000	GPIO111	604	00000010	4
GPIO112	0128	00000000	GPIO112	608	00000010	4
GPIO113	012C	00000000	GPIO113	60C	00000010	4
GPIO114	0130	00000000	GPIO114	610	00000010	4
GPIO115	0134	00000000	GPIO115	614	00000010	4
GPIO116	0138	00000000	GPIO116	618	00000010	4
GPIO117	013C	00000000	GPIO117	61C	00000010	4
GPIO120	0140	00000000	GPIO120	620	00000010	4
GPIO121	0144	00000000	GPIO121	624	00000000	2
GPIO122	0148	00000000	GPIO122	628	00000000	2
GPIO123	014C	00000000	GPIO123	62C	00000010	4
GPIO124	0150	00000000	GPIO124	630	00000010	4
GPIO125	0154	00000000	GPIO125	634	00000010	4
GPIO126	0158	00000000	GPIO126	638	00000010	4
GPIO127	015C	00000000	GPIO127	63C	00000010	4
GPIO130	0160	00000000	GPIO130	640	00000010	4
GPIO131	0164	00000000	GPIO131	644	00000010	4
GPIO132	0168	00000000	GPIO132	648	00000010	4
GPIO133	016C	00000000	GPIO133	64C	00000010	4
GPIO134	0170	00000000	GPIO134	650	00000010	4
GPIO135	0174	00000000	GPIO135	654	00000010	4
GPIO136	0178	00000000	GPIO136	658	00000010	4
GPIO140	0180	00000000	GPIO140	660	00000010	4
GPIO141	0184	00000000	GPIO141	664	00000010	4
GPIO142	0188	00000000	GPIO142	668	00000010	4
GPIO143	018C	00000000	GPIO143	66C	00000010	4
GPIO144	0190	00000000	GPIO144	670	00000010	4
GPIO145	0194	00000000	GPIO145	674	00000010	4
GPIO146	0198	00000000	GPIO146	678	00000010	4
GPIO147	019C	00000000	GPIO147	67C	00000010	4
GPIO150	01A0	00000000	GPIO150	680	00000010	4
GPIO151	01A4	00000000	GPIO151	684	00000010	4

TABLE 24-2: DMA CONTROLLER DEVICE SELECTION (CONTINUED)

Device Name	Device Number (Note 1)	Controller Source
SMBus 1 Controller	2	Slave
	3	Master
SMBus 2 Controller	4	Slave
	5	Master
QUAD SPI Master Controller	6	Transmit
	7	Receive

Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.

24.6 Power, Clocks and Reset

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

24.6.1 POWER DOMAINS

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

24.6.2 CLOCK INPUTS

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

24.6.3 RESETS

Name	Description
nSYSRST	This reset signal resets all of the registers and logic in this block.
DMA_RESET	This reset is generated if either the nSYSRST is asserted or the SOFT_RESET is asserted.

31.11.1 BC-LINK STATUS REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
7	RESET When this bit is ‘1’the BC_Link Master Interface will be placed in reset and be held in reset until this bit is cleared to ‘0’. Setting RESET to ‘1’ causes the BUSY bit to be set to ‘1’. The BUSY remains set to ‘1’ until the reset operation of the BC Interface is completed, which takes approximately 48 BC clocks. The de-assertion of the BUSY bit on reset will not generate an interrupt, even if the BC_BUSY_CLR_INT_EN bit is ‘1’. The BUSY bit must be polled in order to determine when the reset operation has completed.	R/W	1h	nSYSRST
6	BC_ERR This bit indicates that a BC Bus Error has occurred. If an error occurs this bit is set by hardware when the BUSY bit is cleared. This bit is cleared when written with a ’1’. An interrupt is generated If this bit is ‘1’ and BC_ERR_INT_EN bit is ‘1’. Errors that cause this interrupt are: • Bad Data received by the BASE (CRC Error) • Time-out caused by the COMPANION not responding. All COMPANION errors cause the COMPANION to abort the operation and the BASE to time-out.31.11.2	R/WC	0h	nSYSRST
5	BC_ERR_INT_EN This bit is an enable for generating an interrupt when the BC_ERR bit is set by hardware. When this bit is ‘1’, the interrupt signal is enabled. When this bit is ‘0’, the interrupt is disabled.	R/W	0b	nSYSRST
4	BC_BUSY_CLR_INT_EN This bit is an enable for generating an interrupt when the BUSY bit in this register is cleared by hardware. When this bit is set to ‘1’, the interrupt signal is enabled. When the this bit is cleared to ‘0’, the interrupt is disabled. When enabled, the interrupt occurs after a BC Bus read or write.	R/W	0h	nSYSRST
3:1	Reserved	R	-	-
0	BUSY This bit is asserted to ‘1’ when the BC interface is transferring data and on reset. Otherwise it is cleared to ‘0’. When this bit is cleared by hardware, an interrupt is generated if the BC_BUSY_CLR_INT_EN bit is set to ‘1’.	R	1h	nSYSRST

TABLE 37-2: INTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
POWER_UP_EVENT	INPUT	Signal from the RTC/Week Timer block. The POWER_UP_EVENT is asserted by the timer when either the Week_Alarm or the Sub-Week Alarm is asserted. The POWER_UP_EVENT can be suppressed if the SYSPWR_PRES pin indicates that system power is not available.
VTRGD	INPUT	Status signal for the state of the VTR power rail. This signal is high if the power rail is on, and low if the power rail is off.

37.4 Host Interface

The registers defined for the VBAT-Powered Control Interface are accessible only by the EC.

37.5 Power, Clocks and Resets

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

37.5.1 POWER DOMAINS

TABLE 37-3: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR	This power well sources only bus communication. The block continues to operate internally while this rail is down.

37.5.2 CLOCKS

This block does not require clocks.

37.5.3 RESETS

TABLE 37-4: RESET SIGNALS

Name	Description
VBAT_POR	This reset signal is used reset all of the registers and logic in this block.
nSYSRST	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR powered circuitry on-chip. Otherwise it has no effect on the internal state.

43.21 PWM Timing

FIGURE 43-25: PWM OUTPUT TIMING

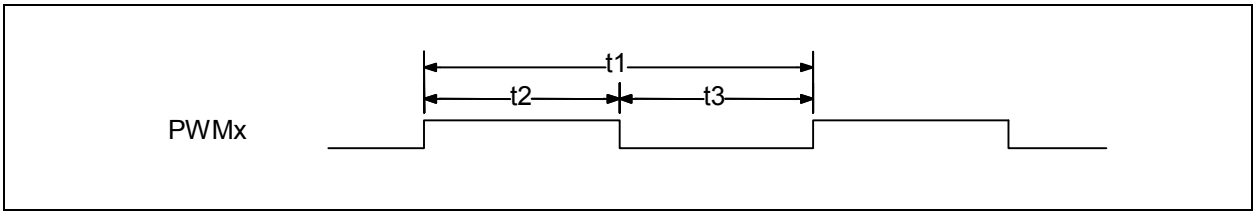

TABLE 43-26: PWM TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	42ns		23.3sec	
t_f	Frequency	0.04Hz		24MHz	
t2	High Time	0		11.65	sec
t3	Low Time	0		11.65	sec
t_d	Duty cycle	0		100	%

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
2460	DMA	0	DMA_CH0_CRC	DMA Channel 0 CRC Enable Register	4
2464	DMA	0	DMA_CH0_CRC	DMA Channel 0 CRC Data Register	4
2468	DMA	0	DMA_CH0_CRC	DMA Channel 0 CRC Post Status Register	4
2480	DMA	0	DMA_CH1	DMA Activate Register	4
2484	DMA	0	DMA_CH1	DMA Memory Start Address Register	4
2488	DMA	0	DMA_CH1	DMA Memory End Address Register	4
248C	DMA	0	DMA_CH1	AHB Address Register	4
2490	DMA	0	DMA_CH1	DMA Control Register	4
2494	DMA	0	DMA_CH1	DMA Channel Interrupt Status	4
2498	DMA	0	DMA_CH1	DMA Channel Interrupt Enable	4
24A0	DMA	0	DMA_CH1_NOCRC	Reserved	22
24C0	DMA	0	DMA_CH2	DMA Activate Register	4
24C4	DMA	0	DMA_CH2	DMA Memory Start Address Register	4
24C8	DMA	0	DMA_CH2	DMA Memory End Address Register	4
24CC	DMA	0	DMA_CH2	AHB Address Register	4
24D0	DMA	0	DMA_CH2	DMA Control Register	4
24D4	DMA	0	DMA_CH2	DMA Channel Interrupt Status	4
24D8	DMA	0	DMA_CH2	DMA Channel Interrupt Enable	4
24E0	DMA	0	DMA_CH2_NOCRC	Reserved	22
2500	DMA	0	DMA_CH3	DMA Activate Register	4
2504	DMA	0	DMA_CH3	DMA Memory Start Address Register	4
2508	DMA	0	DMA_CH3	DMA Memory End Address Register	4
250C	DMA	0	DMA_CH3	AHB Address Register	4
2510	DMA	0	DMA_CH3	DMA Control Register	4
2514	DMA	0	DMA_CH3	DMA Channel Interrupt Status	4
2518	DMA	0	DMA_CH3	DMA Channel Interrupt Enable	4
2520	DMA	0	DMA_CH3_NOCRC	Reserved	22
2540	DMA	0	DMA_CH4	DMA Activate Register	4
2544	DMA	0	DMA_CH4	DMA Memory Start Address Register	4
2548	DMA	0	DMA_CH4	DMA Memory End Address Register	4
254C	DMA	0	DMA_CH4	AHB Address Register	4

TABLE 44-1: REGISTER MEMORY MAP (CONTINUED)

Addr. (Hex)	HW Block Instance Name	HW Block Instance No.	Reg. Bank Name	Reg. Instance Name	Size (Bytes)
5424	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Tx Buffer	4
5430	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 0	4
5434	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 1	4
5438	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 2	4
543C	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 3	4
5440	Quad SPI Master Controller	0	Quad SPI Master Registers	QMSPI Description Buffer 4	4
5800	PWM	0	PWM_EC_Only	PWM Counter ON Time Register	4
5804	PWM	0	PWM_EC_Only	PWM Counter OFF Time Register	4
5808	PWM	0	PWM_EC_Only	PWM Configuration Register	4
580C	PWM	0	PWM_EC_Only	Reserved	4
5810	PWM	1	PWM_EC_Only	PWM Counter ON Time Register	4
5814	PWM	1	PWM_EC_Only	PWM Counter OFF Time Register	4
5818	PWM	1	PWM_EC_Only	PWM Configuration Register	4
581C	PWM	1	PWM_EC_Only	Reserved	4
5820	PWM	2	PWM_EC_Only	PWM Counter ON Time Register	4
5824	PWM	2	PWM_EC_Only	PWM Counter OFF Time Register	4
5828	PWM	2	PWM_EC_Only	PWM Configuration Register	4
582C	PWM	2	PWM_EC_Only	Reserved	4
5830	PWM	3	PWM_EC_Only	PWM Counter ON Time Register	4
5834	PWM	3	PWM_EC_Only	PWM Counter OFF Time Register	4
5838	PWM	3	PWM_EC_Only	PWM Configuration Register	4
583C	PWM	3	PWM_EC_Only	Reserved	4
5840	PWM	4	PWM_EC_Only	PWM Counter ON Time Register	4
5844	PWM	4	PWM_EC_Only	PWM Counter OFF Time Register	4
5848	PWM	4	PWM_EC_Only	PWM Configuration Register	4
584C	PWM	4	PWM_EC_Only	Reserved	4
5850	PWM	5	PWM_EC_Only	PWM Counter ON Time Register	4