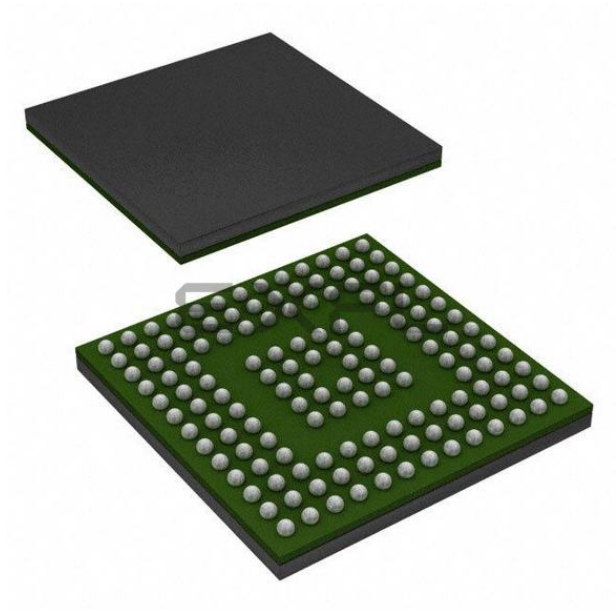


Welcome to E-XFL.COM

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	ARM® Cortex®-M4
Program Memory Type	-
Controller Series	MEC170x
RAM Size	480KB
Interface	ACPI, EBI/EMI, eSPI, I²C, LPC, PECI, PS/2, QSPI, SPI
Number of I/O	123
Voltage - Supply	1.71V ~ 3.465V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	144-WFBGA
Supplier Device Package	144-WFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/mec1704q-c1-i-sz

Signal	Emulated Power Rail	Gated State	Notes
PVT_CS#	VTR	High	
PVT_IO0	VTR	Low	
PVT_IO1	VTR	Low	
PVT_IO2	VTR	Low	
PVT_IO3	VTR	Low	
PWM0	VTR	Low	
PWM1	VTR	Low	
PWM2	VTR	Low	
PWM3	VTR	Low	
PWM4	VTR	Low	
PWM5	VTR	Low	
PWM6	VTR	Low	
PWM7	VTR	Low	
PWM8	VTR	Low	
PWM9	VTR	Low	
PWM10	VTR	Low	
PWROK	VTR	High	
RC_ID0	VTR	Low	
RC_ID1	VTR	Low	
RC_ID2	VTR	Low	
RESETI#	VTR	High	
RESETO#	VTR	High	
SB-TSI_CLK	VTR	High	
SB-TSI_DAT	VTR	High	
SER_IRQ	VTR	Low	Note 2
SHD_CLK	VTR	Low	
SHD_CS#	VTR	High	
SHD_IO0	VTR	Low	
SHD_IO1	VTR	Low	
SHD_IO2	VTR	Low	
SHD_IO3	VTR	Low	
SPI0_CLK	VTR	Low	
SPI0_CS#	VTR	High	
SPI0_MISO	VTR	Low	
SPI0_MOSI	VTR	Low	
SPI1_CLK	VTR	Low	
SPI1_CS#	VTR	High	
SPI1_MOSI	VTR	Low	
SPI1_MSIO	VTR	Low	
TFCLK	VTR	Low	
TFDATA	VTR	Low	
TIN0	VTR	Low	
TIN1	VTR	Low	

MEC1701/MEC1703-128 WFBGA-TF	MEC1704-144 WFBGA-SZ	MEC1705-144 WFBGA-SZ	MEC1701/MEC1703-144 WFBGA-SZ	MEC1701-169 WFBGA-TN	MEC1703-169 WFBGA-TN	MEC1703-169 WFBGA-XY	Interface		Notes
A8	A8	A8	A8	D7	C8	G8	I2C09_SDA	SMB-I2C Controller Port 9 Data	Note 14
M9	N7	N7	N7	N10	N7	R10	I2C10_SCL	SMB-I2C Controller Port 10 Clock	Note 14
J6	N8	N8	N8	J8	N5	P9	I2C10_SDA	SMB-I2C Controller Port 10 Data	Note 14
H10	J13	J13	J13	H11	H10	J14	SB-TSI_CLK	SMB-I2C Controller AMD-TSI Port Clock	
J10	J12	J12	J12	H10	H9	J15	SB-TSI_DAT	SMB-I2C Controller AMD-TSI Port Data	
JTAG and Debug Interface									
B7	C7	C7	C7	C7	B8	D9	JTAG_CLK	JTAG Test Clock. Also ARM SWDCLK	
F12	G11	G11	G11	F10	F12	G11	JTAG_RST#	JTAG Test Reset (active low)	Note 5
A8	A8	A8	A8	D7	C8	G8	JTAG_TDI	JTAG Test Data In	
A7	G8	G8	G8	F8	D8	B9	JTAG_TDO	JTAG Test Data Out. Also ARM SWO	
D9	F8	F8	F8	E7	E7	A9	JTAG_TMS	JTAG Test Mode Select. Also ARM SWDIO	
G10	G13	G13	G12	F11	G11	H12	TFCLK	Trace FIFO debug port - clock	
G9	G12	G12	G9	E11	G9	H9	TFDATA	Trace FIFO debug port - data	
B8	A9	A9	A9	B7	A8	A10	TRACECLK	ARM Embedded Trace Macro Clock	
A6	B7	B7	B7	A7	A7	B8	TRACEDAT0	ARM Embedded Trace Macro Data 0	
B6	F7	F7	F7	F7	A5	D8	TRACEDAT1	ARM Embedded Trace Macro Data 1	
A5	A7	A7	A7	A6	B7	A8	TRACEDAT2	ARM Embedded Trace Macro Data 2	
D8	E7	E7	E7	D4	B5	E8	TRACEDAT3	ARM Embedded Trace Macro Data 3	
Keyboard Scan Interface									
H9	J8	J8	J8	J9	N10	K11	KSI0	Keyboard Scan Matrix Input 0	
K6	L7	L7	L7	L9	M6	R9	KSI1	Keyboard Scan Matrix Input 1	
N9	N9	N9	N9	M9	M8	P10	KSI2	Keyboard Scan Matrix Input 2	
N13	N13	N13	N13	L11	M12	P12	KSI3	Keyboard Scan Matrix Input 3	
K13	K11	K11	K11	F13	L13	P15	KSI4	Keyboard Scan Matrix Input 4	
K12	J9	J9	J9	J13	J9	N14	KSI5	Keyboard Scan Matrix Input 5	
N11	M11	M11	M11	M13	M11	M11	KSI6	Keyboard Scan Matrix Input 6	
J9	L10	L10	L10	K9	L10	M10	KSI7	Keyboard Scan Matrix Input 7	
M11	M10	M10	M10	M11	N11	L11	KSO00	Keyboard Scan Matrix Output 0	
A9	E8	E8	E8	E8	E8	A11	KSO01	Keyboard Scan Matrix Output 1	
E10	E9	E9	E9	E10	F11	F10	KSO02	Keyboard Scan Matrix Output 2	
B13	F13	F13	F13	A12	D13	E15	KSO03	Keyboard Scan Matrix Output 3	
K10	L13	L13	L13	H13	J10	M14	KSO04	Keyboard Scan Matrix Output 4	
J12	K13	K13	K13	J12	J12	K15	KSO05	Keyboard Scan Matrix Output 5	

Agg IRQ	Agg Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description	Agg NVIC	Direct NVIC
GIRQ12	0	GPIO200	GPIO Event	Yes	GPIO Interrupt Event	4	N/A
	1	GPIO201	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO202	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO203	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO204	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO205	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO206	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO207	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO210	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO211	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO212	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO213	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO214	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO215	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO216	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO217	GPIO Event	Yes	GPIO Interrupt Event		
	16	Reserved	-	-			
	17	GPIO221	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO222	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO223	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO224	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO225	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO226	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO227	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO230	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO231	GPIO Event	Yes	GPIO Interrupt Event		
	26	Reserved	-	-			
	27	GPIO233	GPIO Event	Yes	GPIO Interrupt Event		
	28	GPIO234	GPIO Event	Yes	GPIO Interrupt Event		
	29	Reserved	-	-			
	30	Reserved	-	-			
	31	Reserved					
GIRQ13	0	SMB-I2C Controller 0	SMB-I2C	No	SMB-I2C Controller 0 Interrupt Event	5	20
	1	SMB-I2C Controller 1	SMB-I2C	No	SMB-I2C Controller 1 Interrupt Event		21
	2	SMB-I2C Controller 2	SMB-I2C	No	SMB-I2C Controller 2 Interrupt Event		22
	3	SMB-I2C Controller 3	SMB-I2C	No	SMB-I2C Controller 3 Interrupt Event		23
	4-31	Reserved					

Agg IRQ	Agg Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description	Agg NVIC	Direct NVIC
GIRQ18	0	LPC	LPC_INTERNAL_ ERROR	No	LPC BAR conflict or internal bus error	10	90
	1	Quad Master SPI Controller	QMSPI_INT	No	Master SPI Controller Requires Servicing		91
	2	GP-SPI 0	TXBE_STS	No	SPI TX buffer empty		92
	3	GP-SPI 0	RXBF_STS	No	SPI RX buffer full		93
	4	GP-SPI 1	TXBE_STS	No	SPI TX buffer empty		94
	5	GP-SPI 1	RXBF_STS	No	SPI RX buffer full		95
	6	BC-Link 0	BCM_BUSY_CLR	No	BC-Link Busy Clear Flag		97
	7	BC-Link 0	BCM_ERR	No	BC-Link Error Flag Interrupt		96
	8	BC-Link 1	BCM_BUSY_CLR	No	BC-Link Busy Clear Flag		99
	9	BC-Link 1	BCM_ERR	No	BC-Link Error Flag Interrupt		98
	10	PS2 Interface 0	PS2_ACT	No	PS/2 Device Interface 0 - Activity Interrupt Event		100
	11	PS2 Interface 1	PS2_ACT	No	PS/2 Device Interface 1 - Activity Interrupt Event		101
	12	PS2 Interface 2	PS2_ACT	No	PS/2 Device Interface 2 - Activity Interrupt Event		102
	13	EEPROM	EEPROM	No	EEPROM Transfer Complete		155
	14-31	Reserved					
GIRQ19	0	eSPI_Slave	INTR_PC	No	Peripheral Channel Interrupt	11	103
	1	eSPI_Slave	INTR_BM1	No	Bus Mastering Channel 1 Interrupt		104
	2	eSPI_Slave	INTR_BM2	No	Bus Mastering Channel 2 Interrupt		105
	3	eSPI_Slave	INTR_LTR	No	Peripheral Message (LTR) Interrupt		106
	4	eSPI_Slave	INTR_OOB_ UP	No	Out of Band Channel Up Interrupt		107
	5	eSPI_Slave	INTR_OOB_ DOWN	No	Out of Band Channel Down Interrupt		108
	6	eSPI_Slave	INTR_FLASH	No	Flash Channel Interrupt		109
	7	eSPI_Slave	eSPI_RESET	No	eSPI_RESET		110
	8	eSPI_Slave	VWIRE_ENABLE	No	Virtual Wire Channel Enable Asserted		156
	9-31	Reserved					
GIRQ20	0-8	Test	Test	-	-	12	N/A
	9-31	Reserved					

5.7.2 NVIC RELATIONSHIP TO EXCEPTION VECTOR TABLE ENTRIES

The Vector Table consists of 4-byte entries, one per vector. Entry 0 is not a vector, but provides an initial Reset value for the Main Stack Pointer. Vectors start with the Reset vector, at Entry #1. Entries up through #15 are dedicated for internal exceptions, and do not involve the NVIC.

NVIC entries in the Vector Table start with Entry #16, so that NVIC Interrupt #0 is at Entry #16, and all NVIC interrupt numbers are incremented by 16 before accessing the Vector Table.

The number of connections to the NVIC determines the necessary minimum size of the Vector Table, as shown below. It can extend as far as 256 entries (255 vectors, plus the non-vector entry #0).

A Vector entry is used to load the Program Counter (PC) and the EPSR.T bit. Since the Program Counter only expresses code addresses in units of two-byte Halfwords, bit[0] of the vector location is used to load the EPSR.T bit instead, selecting THUMB mode for exception handling. Bit[0] must be '1' in all vectors, otherwise a UsageFault exception will be posted (INVSTATE, unimplemented instruction set). If the Reset vector is at fault, the exception posted will be HardFault instead.

TABLE 5-4: EXCEPTION AND INTERRUPT VECTOR TABLE LAYOUT

Table Entry	Exception Number	Exception
Special Entry for Reset Stack Pointer		
0	(none)	Holds Reset Value for the Main Stack Pointer. Not a Vector.
Core Internal Exception Vectors start here		
1	1	Reset Vector (PC + EPSR.T bit)
2	2	NMI (Non-Maskable Interrupt) Vector
3	3	HardFault Vector
4	4	MemManage Vector
5	5	BusFault Vector
6	6	UsageFault Vector
7	(none)	(Reserved by ARM Ltd.)
8	(none)	(Reserved by ARM Ltd.)
9	(none)	(Reserved by ARM Ltd.)
10	(none)	(Reserved by ARM Ltd.)
11	11	SVCall Vector
12	12	Debug Monitor Vector
13	(none)	(Reserved by ARM Ltd.)
14	14	PendSV Vector
15	15	SysTick Vector
NVIC Interrupt Vectors start here		
16	16	NVIC Interrupt #0 Vector
. . .	. . .	. . .
n + 16	n + 16	NVIC Interrupt #n Vector
. . .	. . .	. . .
max + 16	max + 16	NVIC Interrupt #max Vector (Highest-numbered NVIC connection.)
. . .	. . .	. Table size may (but need not) extend further. . .
255	255	NVIC Interrupt #239 (Architectural Limit of Exception Table)

place in parallel with the data transfer; enabling CRC will not increase the time to complete a DMA transaction. The CRC generator has the optional ability to automatically transfer the generated CRC to the destination after the data transfer has completed.

CRC generation is subject to a number of restrictions:

- The CRC is only generated on channels that have the CRC hardware. See Table 7-10, "Channel Register Summary" for a definition of which channels have the ability to generate a CRC
- The DMA transfer must be 32-bits
- If CRC is enabled, DMA interrupts are inhibited until the CRC is completed, including the optional post-transfer copy of it is enabled
- The CRC must be initialized by firmware. The value FFFFFFFFh must be written to the Data Register in order to initialize the generator for the standard CRC-32-IEEE algorithm
- The CRC will bit-order reverse In and Out, and Invert Out, as required by the CRC algorithm

7.8.2.5 Block Fill Option

A Fill engine can be attached to a DMA channel in order to provide a fast mechanism to set a block of memory to a fixed value (for example, clearing a block of memory to zero). The block fill operation runs approximately twice as fast as a memory-to-memory copy.

In order to fill memory with a constant value, firmware **must** configure the channel in the following order:

1. Set the DMA Channel N Fill Data Register to the desired fill value
2. Set the DMA Channel N Fill Enable Register to ‘1b’, enabling the Fill engine
3. Set the DMA Channel N Control Register to the following values:
   - RUN = 0
   - TRANSFER_DIRECTION = 0 (memory destination)
   - INCREMENT_MEMORY_ADDRESS = 1 (increment memory address after each transfer)
   - INCREMENT_DEVICE_ADDRESS = 1
   - DISABLE_HARDWARE_FLOW_CONTROL = 1 (no hardware flow control)
   - TRANSFER_SIZE = 1, 2 or 4 (as required)
   - TRANSFER_ABORT = 0
   - TRANSFER_GO = 1 (this starts the transfer)

7.9 EC Registers

The DMA Controller consists of a Main Block and a number of Channels. Table 7-9, "Main Register Summary" lists the registers in the Main Block and Table 7-10, "Channel Register Summary" lists the registers in each channel. Addresses for each register are determined by adding the offset to the Base Address for the DMA Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Registers are listed separately for the Main Block of the DMA Controller and for a DMA Channel. Each Channel has the same set of registers. The absolute register address for registers in each channel are defined by adding the Base Address for the DMA Controller Block, the Offset for the Channel shown in Table 7-8, "DMA Channel Offsets" to the offsets listed in Table 7-9, "Main Register Summary" or Table 7-10, "Channel Register Summary".

TABLE 7-8: DMA CHANNEL OFFSETS

Instance Name	Channel Number	Offset
DMA Controller	Main Block	000h
DMA Controller	0	040h
DMA Controller	1	080h
DMA Controller	2	0C0h
DMA Controller	3	100h
DMA Controller	4	140h
DMA Controller	5	180h
DMA Controller	6	1C0h
DMA Controller	7	200h

TABLE 9-10: I/O BASE ADDRESS REGISTER DEFAULT VALUES (CONTINUED)

LPC Index	EC Offset	Logical Device	Reset Default	Default LPC I/O Address	Valid	LDN	Mask
98h	398h	Embedded Memory Interface (EMI) 2	0000_120Fh	0000h	0h	12h	Fh
9Ch	39Ch	BIOS Debug Port (Port 80) 0	0000_2000h	0000h	0h	20h	0h
A0h	3A0h	BIOS Debug Port (Port 80) 1	0000_2100h	0000h	0h	21h	0h
A4h	3A4h	RTC	0000_141Fh	0000h	0h	14h	1Fh

9.9.4 SRAM BASE ADDRESS REGISTERS

9.9.4.1 SRAM Base Address Register Format, LPC Configuration Register Format

Offset	See Table 9-11, "SRAM Base Address Register Default Values, LPC Config"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
63:32	LPC_HOST_ADDRESS These 32 bits are used to match LPC Memory addresses	R/W	0h	RESET_HOST
31:8	Reserved	R	-	-
7	VALID If this bit is 1, this SRAM Memory BAR is valid and will participate in LPC matches. If it is 0 this BAR is ignored.	R/W	0h	RESET_HOST
6:0	Reserved	R	-	-

9.9.4.2 SRAM Base Address Register Format, EC-Only Register Format

Offset	See Table 9-12, "SRAM Base Address Register Default Values, EC-Only"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	AHB_BASE These 24 bits define the base of a region in AHB address space that will be mapped to the LPC Memory space. Valid AHB addresses are integer multiples of the memory size. For example, if the memory is 4k bytes than the AHB Base address must be located on a 4k byte boundary. The 24 bits in this field are left-shifted by 8 bits to form a 32-bit AHB address, so all memory blocks begin on a 256-byte boundary.	R/W	0h	RESET_SYS
7	INHIBIT Host access to the memory block is inhibited when this bit is 1. The Host can access the memory region mapped by the fields AHB Base and Size when this bit is 0.	R/W	0h	RESET_SYS
6:4	Reserved	R	-	-

13.13.1 OS2EC DATA EC BYTE 0 REGISTER

This register is aliased; see Section 13.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 13.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 13.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS_TO_EC_DATA_BYTE_0 This is byte 0 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	RESET_SYS

13.13.1.1 OS2EC DATA BYTES[3:0]

When the CMD bit in the OS STATUS OS Register is cleared to '0', reads by the ACPI_EC from the OS2EC DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0].

All access to the OS2EC DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is cleared to '0', the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the OS STATUS OS Register.
2. Reads from the OS2EC Data EC Byte 0 Register clears the IBF bit in the OS STATUS OS Register.
3. All reads from OS2EC DATA BYTES[3:1] return 00h without error.
4. Access to OS2EC DATA BYTES[3:1 has no effect on the IBF and OBF bits in the OS STATUS OS Register.

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is set to '1', the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the OS STATUS OS Register.
2. Reads from the OS2EC Data EC Byte 3 Register clears the IBF bit in the OS STATUS OS Register.

13.13.2 OS2EC DATA EC BYTE 1 REGISTER

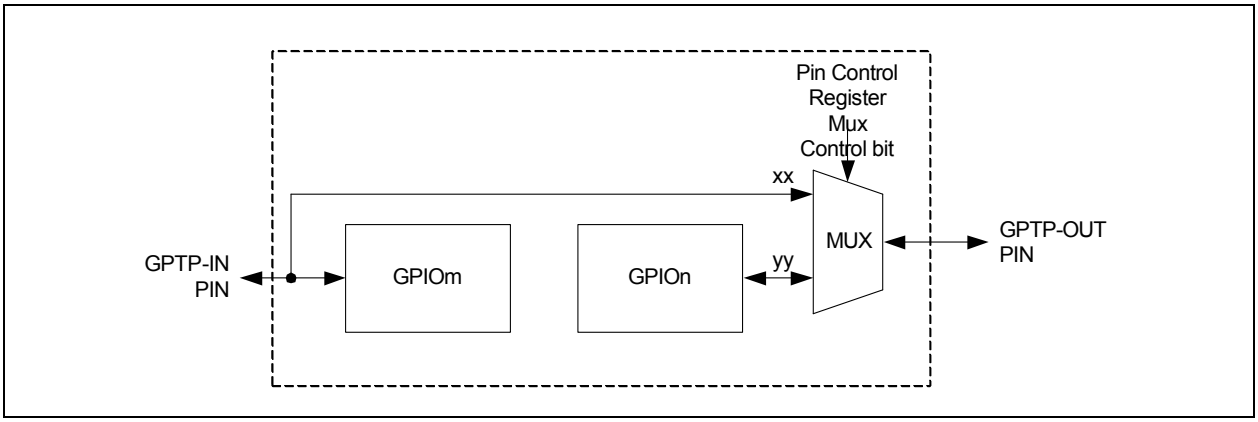
This register is aliased; see Section 13.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 13.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 13.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	109h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_ BYTE_1 This is byte 1 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	RESET_SYS

13.13.3 OS2EC DATA EC BYTE 2 REGISTER

This register is aliased; see Section 13.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 13.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 13.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

FIGURE 18-2: GPIO PASS-THROUGH PORT EXAMPLE



The Pin Control Register Mux Control fields shown in Figure 18-2 are illustrated as ‘xx’ and ‘yy’ because this figure is an example, it does not represent the actual GPIO multiplexing configuration. The GPIO Multiplexing tables in this chapter must be used to determine the correct values to use to select between a GPIO and the pass-through.

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.

18.6 Accessing GPIOs

There are two ways to access GPIO output data. GPIO_OUTPUT_SELECT is used to determine which GPIO output data bit affects the GPIO output pin.

- Group Output GPIO Data
  - Outputs to individual GPIO ports are grouped into 32-bit GPIO Output Registers.
- Individual Output GPIO Data
  - Each GPIO output port may be individually accessible in the ALTERNATE_GPIO_DATA field of the port’s Pin Control Register. On reads, ALTERNATE_GPIO_DATA returns the programmed value, not the value on the pin.

There are two ways to access GPIO input data.

- Group Input GPIO Data
  - Inputs from individual GPIO ports are grouped into 32-bit GPIO Input Registers and always reflect the current state of the GPIO input from the pads, independent of the setting of the MUX_CONTROL field in the Pin Control Register.
- Group Input GPIO Data
  - Each GPIO input port is individually accessible in the GPIO_INPUT field of the port’s Pin Control Register. The GPIO_INPUT field always reflects the current state of GPIO input from the pad, independent of the setting of the MUX_CONTROL field in the Pin Control Register.

18.6.1 HOST ACCESS OF GPIOS

The GPIO Output Registers and GPIO Input Registers can be configured to be Host accessible via one of the SRAM Base Address Register, if the base of the internal address of the BAR is set to an offset of 200h from the GPIO base address, with a SIZE of 9, setting of block of 512 bytes. All of the Output and Input registers can then be accessed as offsets from the Host base address.

The GPIO Output and Input registers can also be accessed as one of the regions in an EMI block. This access is defined in the EMI Protocols chapter of the firmware specification.

18.7 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 MEC170x GPIO indexing is done sequentially starting from GPIO000.

25.11.11 REGISTER A

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UPDATE_IN_PROGRESS '0' indicates that the Time and Date registers are stable and will not be altered by hardware soon. '1' indicates that a hardware update of the Time and Date registers may be in progress, and those registers should not be accessed by the host program. This bit is set to '1' at a point 488us (16 cycles of the 32K clock) before the update occurs, and is cleared immediately after the update. See also the Update-Ended Interrupt, which provides more useful status.	R	0b	RESET_RTC
6:4	DIVISION_CHAIN_SELECT This field provides general control for the Time and Date register updating logic. 11xb=Halt counting. The next time that 010b is written, updates will begin 500ms later. 010b=Required setting for normal operation. It is also necessary to set the Block Enable bit in the RTC Control Register to '1' for counting to begin 000b=Reserved. This field should be initialized to another value before Enabling the block in the RTC Control Register Other values Reserved	R/W	000b	RESET_RTC
3:0	RATE_SELECT This field selects the rate of the Periodic Interrupt source. See Table 25-7	R/W	0h	RESET_RTC

TABLE 25-7: REGISTER A FIELD RS: PERIODIC INTERRUPT SETTINGS

RS (hex)	Interrupt Period
0	Never Triggered
1	3.90625 ms
2	7.8125 ms
3	122.070 us
4	244.141 us
5	488.281 us
6	976.5625 us
7	1.953125 ms
8	3.90625 ms
9	7.8125 ms
A	15.625 ms
B	31.25 ms
C	62.5 ms
D	125 ms
E	250 ms
F	500 ms

Note 1: The Week Alarm Counter **must not** be modified by firmware if Sub-Week Alarm Counter is using the Week Alarm Counter as its clock source (i.e., the SUBWEEK_TICK field is set to any of the values 4, 5, 6 or 7). The Sub-Week Alarm Counter must be disabled before changing the Week Alarm Counter. For example, the following sequence may be used:

1.Write 0h to the Sub-Week Alarm Counter Register (disabling the Sub-Week Counter)

2.Write the Week Alarm Counter Register

3.Write a new value to the Sub-Week Alarm Counter Register, restarting the Sub-Week Counter

26.9.1.3 15-bit Clock Divider

This counter is 15 bits wide. The clock for this counter is 32KHz, and as long as the Week Timer is enabled, it is incremented at 32.768KHz rate. The Clock Divider automatically The Clock Divider generates a clock out of 1 Hz when the counter wraps from 7FFFh to 0h.

By selecting one of the 15 bits of the counter, using the Sub-Second Programmable Interrupt Select Register, the Clock Divider can be used either to generate a time base for the Sub-Week Alarm Counter or as an isochronous interrupt to the EC, the SUB_SECOND interrupt. See Table 26-9, "SPISR Encoding" for a list of available frequencies.

26.9.2 TIMER VALID STATUS

If power on reset occurs on the VBAT power rail while the main device power is off, the counters in the Week Alarm are invalid. If firmware detects a POR on the VBAT power rail after a system boot, by checking the status bits in the Power, Clocks and Resets registers, the Week Alarm block must be reinitialized.

26.9.3 APPLICATION NOTE: REGISTER TIMING

Register writes in the Week Alarm complete within two cycles of the 32KHz clock.The write completes even if the main system clock is stopped before the two cycles of the 32K clock complete. Register reads complete in one cycle of the internal bus clock.

All Week Alarm interrupts that are asserted within the same cycle of the 32KHz clock are synchronously asserted to the EC.

26.9.4 APPLICATION NOTE: USE OF THE WEEK TIMER AS A 43-BIT COUNTER

The Week Timer cannot be directly used as a 42-bit counter that is incremented directly by the 32.768KHz clock domain. The upper 28 bits (28-bit Week Alarm Counter) are incremented at a 1Hz rate and the lower 16 bits (15-bit Clock Divider) are incremented at a 32.768KHz rate, but the increments are not performed in parallel. In particular, the upper 28 bits are incremented when the lower 15 bits increment from 0 to 1, so as long as the Clock Divider Register is 0 the two registers together, treated as a single value, have a smaller value then before the lower register rolled over from 7FFFh to 0h.

The following code can be used to treat the two registers as a single large counter. This example extracts a 32-bit value from the middle of the 43-bit counter:

```
dword TIME_STAMP(void)
{
    AHB_dword wct_value;
    AHB_dword cd_value1;
    AHB_dword cd_value2;
    dword irqEnableSave;

    //Disable interrupts
    irqEnableSave = IRQ_ENABLE;
    IRQ_ENABLE = 0;

    //Read 15-bit clk divider reading register, save result in A
    cd_value1 = WTIMER->CLOCK_DIVIDER;
    //Read 28 bit up-counter timer register, save result in B
    wct_value = WTIMER->WEEK_COUNTER_TIMER;
    //Read 15-bit clk divider reading register, save result in C
    cd_value2 = WTIMER->CLOCK_DIVIDER;

    if (0 == cd_value2)
```

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	WEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Week Alarm Counter Register is greater than or equal the contents of the Week Timer Compare Register and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit does not have to be cleared to remove a Week Timer Power-Up Event.	R/WC	0	RESET_VBAT
0	SUBWEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Sub-Week Alarm Counter Register decrements from '1' to '0' and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit MUST be cleared to remove a Sub-Week Timer Power-Up Event.	R/WC	0	RESET_VBAT

26.11.7 SUB-WEEK ALARM COUNTER REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	Reserved	R	-	-
24:16	SUBWEEK_COUNTER_STATUS Reads of this register return the current state of the 9-bit Sub-Week Alarm counter.	R	00h	RESET_VBAT
15:9	Reserved	R	-	-
8:0	SUBWEEK_COUNTER_LOAD Writes with a non-zero value to this field reload the 9-bit Sub-Week Alarm counter. Writes of 0 disable the counter. If the Sub-Week Alarm counter decrements to 0 and the AUTO_RELOAD bit is set, the value in this field is automatically loaded into the Sub-Week Alarm counter.	R/W	00h	RESET_VBAT

30.6 Host Interface

The registers defined for the ADC are accessible by the various hosts as indicated in Section 30.11, "EC Registers".

30.7 Power, Clocks and Reset

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

30.7.1 POWER DOMAINS

TABLE 30-2: POWER SOURCES

Name	Description
VTR	This power well supplies power for the registers tn this block.
VTR_ANALOG	This power well supplies power for the analog circuitry in this block.

30.7.2 CLOCK INPUTS

TABLE 30-3: CLOCK INPUTS

Name	Description
16MHz	This derived clock signal drives controls the conversion rate of the ADC. At 16MHz, the ADC does one channel conversion in 1.125µS.

30.7.3 RESETS

TABLE 30-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.

30.8 Interrupts

TABLE 30-5: EC INTERRUPTS

Source	Description
ADC_Single_Int	Interrupt signal from ADC controller to EC for Single-Sample ADC conversion.
ADC_Repeat_Int	Interrupt signal from ADC controller to EC for Repeated ADC conversion.

30.9 Low Power Modes

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

The ADC is designed to conserve power when it is either sleeping or disabled. It 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.'

- As long as START_REPEAT is 1 the ADC will repeatedly begin conversion cycles with a period defined by REPEAT_DELAY.
- If the delay period expires and a conversion cycle is already in progress because START_SINGLE was written with a 1, the cycle in progress will complete, followed immediately by a conversion cycle using RPT_EN to control the channel conversions.

30.10.2 SINGLE MODE

- The Single Mode conversion cycle will begin without a delay. After all channels enabled by SINGLE_EN are complete, SINGLE_DONE_STATUS will be set to 1. When the next conversion cycle begins the bit is cleared.
- If START_SINGLE 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 to control the channel conversions.

30.10.3 APPLICATION NOTES

Transitions on ADC GPIOs are not permitted when Analog to Digital Converter readings are being taken.

Note:	ADC inputs require at least a 0.1 uF capacitor to filter glitches. See the MEC170x PCB Layout Guide for ADC filtering recommendations.

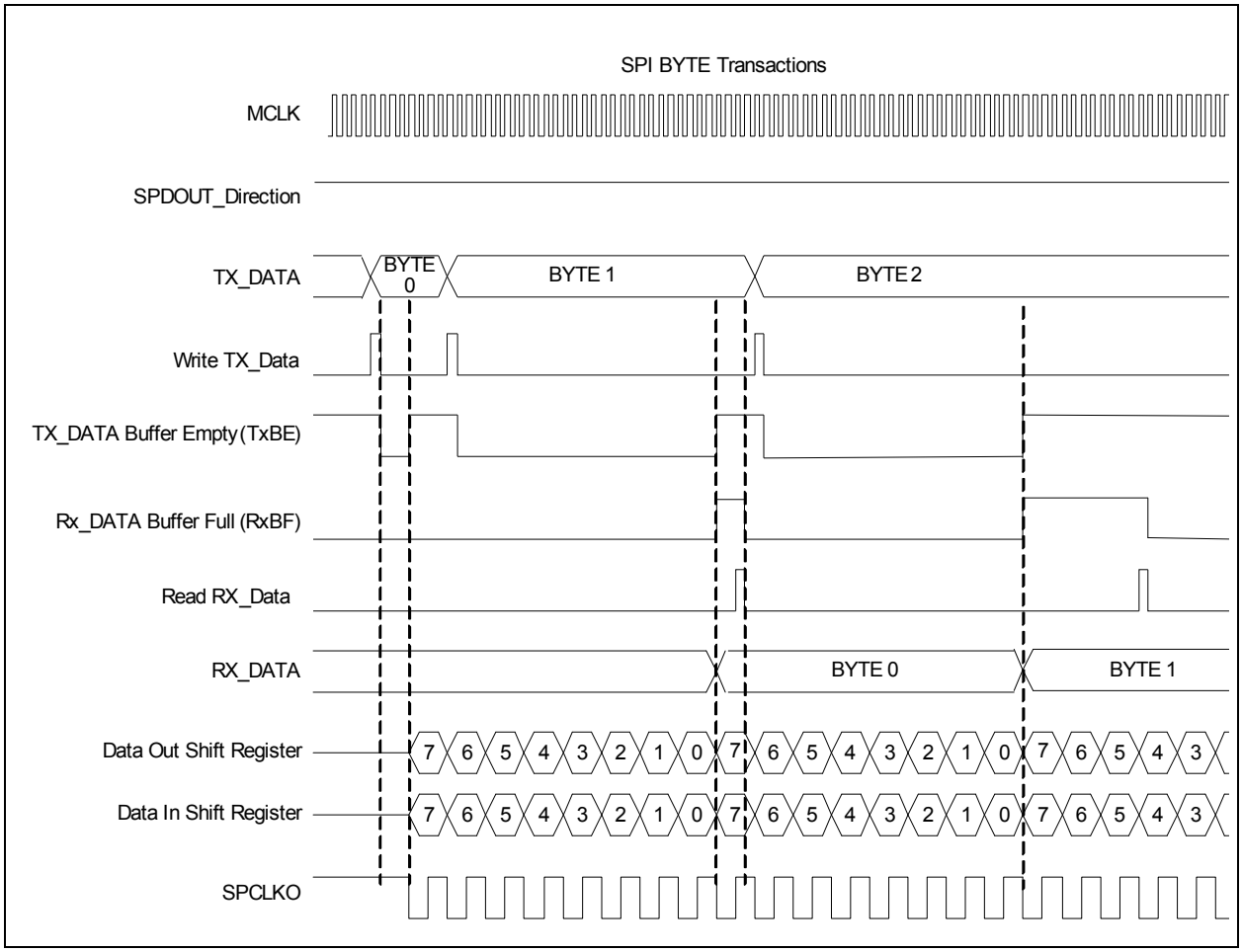
30.11 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Analog to Digital Converter Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 30-6: REGISTER SUMMARY

Offset	Register Name
00h	ADC Control Register
04h	ADC Delay Register
08h	ADC Status Register
0Ch	ADC Single Register
10h	ADC Repeat Register
14h	ADC Channel 0 Reading Register
18h	ADC Channel 1 Reading Register
1Ch	ADC Channel 2 Reading Register
20h	ADC Channel 3 Reading Register
24h	ADC Channel 4 Reading Register
28h	ADC Channel 5 Reading Register
2Ch	ADC Channel 6 Reading Register
30h	ADC Channel 7 Reading Register
34h	ADC Channel 8 Reading Register
38h	ADC Channel 9 Reading Register
3Ch	ADC Channel 10 Reading Register
40h	ADC Channel 11 Reading Register
44h	ADC Channel 12 Reading Register
48h	ADC Channel 13 Reading Register
4Ch	ADC Channel 14 Reading Register
50h	ADC Channel 15 Reading Register

FIGURE 37-3: 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 SPI Control Register. The transmit data is shifted out on the edge as selected by the TCLKPH bit in the SPI Clock Control Register. All received data can be sampled on a rising or falling SPI_CLK edge using the RCLKPH bit in the SPI Clock Control Register 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 MEC170x 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 37.10.3, "Types of SPI Transactions".

37.10.2 DMA MODE

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

39.6 Host Interface

The registers defined for the PS/2 Interface are accessible by the various hosts as indicated in Section 39.14, "EC Registers".

39.7 Power, Clocks and Reset

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

39.7.1 POWER DOMAINS

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

39.7.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for PS/2 Interface logic.
2 MHz Clock	The PS/2 state machine is clocked using the 2 MHz clock.

39.7.3 RESETS

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

39.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
PS2_x	Interrupt request to the Interrupt Aggregator for PS2 controller instance *x*, based on PS2 controller activity. Section 39.14.4, "PS2 Status Register" defines the sources for the interrupt request.
PS2_*x*_WK	Wake-up request to the Interrupt Aggregator's wake-up interface for PS2 port *x*. In order to enable PS2 wakeup interrupts, the pin control registers for the PS2DAT pin must be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

39.9 Low Power Modes

The PS/2 Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

The PS2 interface will only sleep while the PS2 is disabled or in Rx mode with no traffic on the bus.

39.10 Description

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 12m, as required by the PS/2 specification. A pull-up resistor, typically 10K, is connected to both lines. This allows either the 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 (PS2CLK and PS2DAT) are driven by open drain drivers which can be pulled to VTR or the main power rail (+3.3V nominal) through 10K-ohm resistors.

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
Note 1: The recommended procedure is to first clear the VTR WDT STATUS bit, increment the WDT_EVENT_COUNT, clear the VBAT WDT STATUS bit and then rearm the WDT.				

46.8.9 PECI DISABLE REGISTER

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	R	-	-
0	PECI_DISABLE This bit reduces leakage current through the CPU voltage reference pin if PECI or SB-TSI are not used. 1=The VREF_VTT function is disabled, independent of the mux setting of the GPIO that shares the pin. The GPIO that shares the pin is not disabled 0=The VREF_VTT pin is enabled	R/W	0b	RESET_SYS

46.8.10 GPIO BANK POWER REGISTER

Offset	64h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	–
7	GPIO Bank Power Lock 0 = VTR_LEVEL bits[2:0] and GPIO Bank Power Lock bit are R/W 1 = VTR_LEVEL bits[2:0] and GPIO Bank Power Lock bit are Read Only This bit cannot be cleared once it is set to ‘1’. Writing zero has no effect.	Bit[7]=0 R/W Bit[7]=1 RO	0h	RESET_SYS
6:3	Reserved	R	-	–
2	VTR_LEVEL3 Voltage value on VTR3. This bit is set by hardware after a VTR Power On Reset, but may be overridden by software. It must be set by software if the VTR power rail is not active when RESET_SYS is de-asserted. 1=VTR3 is powered by 1.8V 0=VTR3 is powered by 3.3V	see Bit[7]	0h	RESET_SYS

7. Set the IP_CS bit in the Manual Control Register to ‘0b’. The disables the eFuse memory array
8. Select the IP_ADDR_HI field in the Manual Mode Address Register to the block number of the block to be programmed
9. Set the IP_CS bit in the Manual Control Register to ‘1b’. The enables and powers up the selected block in the eFuse memory array
10. Select the IP_ADDR_LO field in the Manual Mode Address Register to the address of the bit within the current block to be programmed
11. Wait 100 ns (min)
12. Set PROG_EN to HIGH for 10µs (typical.
13. Set PROG_EN to LOW for 100 ns (min)
14. Repeat steps 10 to 13 for all bits within a block of 1K bits that are to be programmed to ‘1b’
15. In order to program bits in another 1K bit block, go back to Step 7 and follow the subsequent steps to programs the bits in that block

Programming one 1K bit block at a time eliminates glitches when switching between physical eFUSE blocks to prevent memory corruption.

Power down sequence after programming operation

1. Set the IP_CS bit in the Manual Control Register to ‘0b’. The disables and powers down the eFuse memory array
2. Set the VREF_ADC pin to ground
3. Set FSOURCE_EN_READ to ‘1b’
4. Set FSOURCE_EN_PRGM to ‘0b’ - DO NOT combine with step 3; writing FSOURCE_EN_PRGM and FSOURCE_EN_READ MUST be performed with separate writes

After programming is completed, the eFuse memory array may be read.

49.8 eFuse Reading Sequence

After power-up, the eFUSE Block is enabled for reading, but the block is in low-power/disabled mode.

1. Control Register should contain 0x10 (FSOURCE_EN_READ = 1); set this bit if it is not already set.
2. Set the bit 0 to 1b (ENABLE block) in Control Register.
3. Read the desired eFuse content (by reading eFUSE Memory).
4. If power savings is desired, turn the eFuse block off (clear ENABLE bit in Control Register)

49.9 eFuse Memory Map

The eFuse memory array is organized into four regions. Each of the four regions may be locked by a control bit in the EC Register Bank. When locked, a region in eFuse memory cannot be written, and always returns 0 on reads. The lock bits are located in the OTP Lock Register. In the following table, the four regions are identified by the lock bit names in the Lock Register.

TABLE 49-5: EFUSE MEMORY MAP

Byte Number	Lock Bit	Location Name	Description
0-31	PRIVATE_KEY_LOCK	Encryption ECDH private key (aka, ECC private key)	256-bit P-256 Elliptic Curve private key, for key exchange as part of the optional decryption step in the Boot ROM Load process. Stored big-endian. • If this region is programmed by Microchip, it is encrypted with AES-256 and is always locked when the Boot ROM exits. • If this region is not programmed by Microchip, it is not encrypted, and is left unlocked when the Boot ROM exits and can be programmed by customers.
32-127	MCHIP_LOCK	Microchip	OTP data required by Microchip. This region is always locked when the Boot ROM exits.

TABLE 50-11: VTR SUPPLY CURRENT, I_VTR (CONTINUED)

VTR	VCC	System State	48 MHz PLL	EC_CLK Freq	Typical (3.3V, 25^0 C)	Max (3.45V, 70^0 C)	Max (3.45V, 85^0 C) (Note 3)	Units	Comments (Note 1)
On	On	S0	On	Off	2.0	3.0	4.5	mA	Light Sleep, LPC Clock ON (**Note 2**) or eSPI Traffic
On	On	S0	Off	Off	1.0	2.0	3.1	mA	Heavy Sleep, LPC Clock ON (**Note 2**) or eSPI Traffic
On	Off	S5	On	48MHz	12.5	14.5	16.0	mA	FULL ON (48MHz), LPC Clock Off and No eSPI Traffic
On	Off	S5	On	12MHz	8.0	9.5	11.5	mA	FULL ON (12MHz), LPC Clock Off and No eSPI Traffic
On	Off	S5	On	1MHz	5.5	6.5	8.0	mA	FULL ON (1MHz), LPC Clock Off and No eSPI Traffic
On	Off	S5	On	Off	1.5	2.5	4.0	mA	Light Sleep , LPC Clock Off (**Note 2**) and No eSPI Traffic
On	Off	S5	Off	Off	0.5	1.9	3.0	mA	Heavy Sleep, LPC Clock Off (**Note 2**) and No eSPI Traffic

Note 1: FULL ON is defined as follows: The processor is not sleeping, the Core regulator and the PLL remain powered, and at least one block is not sleeping.

2: The sleep states are defined in the System Sleep Control Register in the Power, Clocks and Resets Chapter. See Table 4.9.4, "System Sleep Control Register".

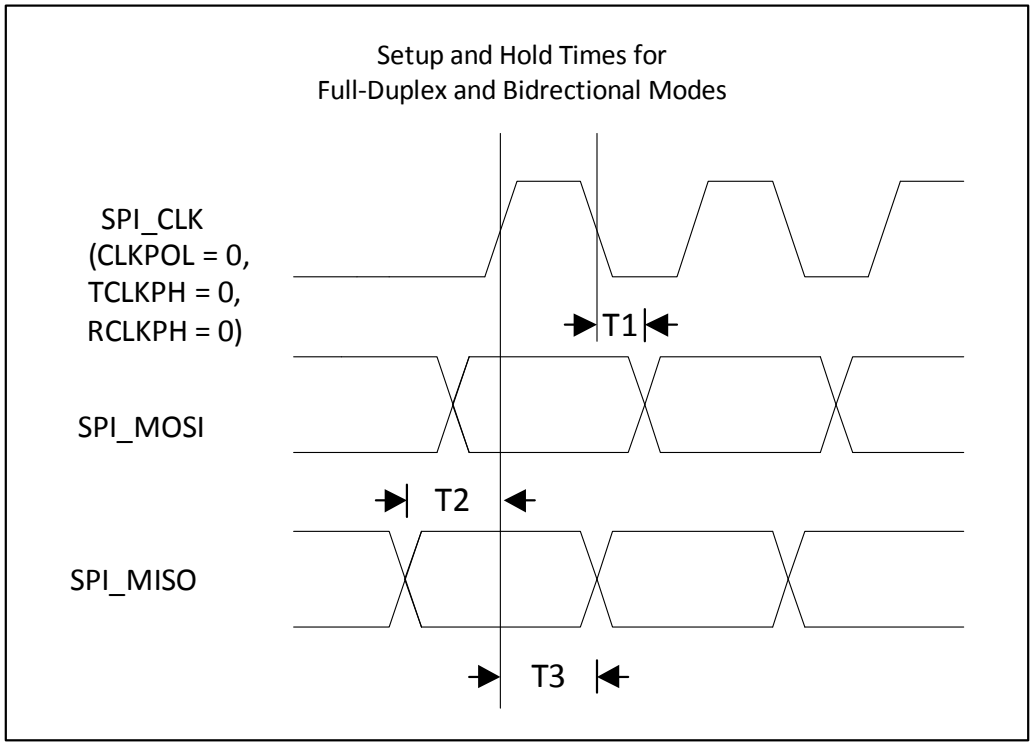
3: Applicable to MEC1705 and MEC1704 only

Note:	In order to achieve the lowest leakage current when the VREF_VTT power domain is not required, ground the VREF_VTT pin.

TABLE 50-12: VBAT SUPPLY CURRENT, I_VBAT (VBAT=3.0V)

VCC	VTR	System State	48 MHz PLL	Typical (3.0V, 25^0 C)	Max (3.0V, 25^0 C)	Units	Comments
Off	Off	S5	Off	11.0	20.0	uA	Internal 32kHz oscillator
Off	Off	S5	Off	5.0	9.0	uA	32kHz crystal oscillator
Off	Off	S5	Off	5.0	9.0	uA	External 32kHz clock on XTAL2 pin

FIGURE 51-30: SPI SETUP AND HOLD TIMES



Note:	SPI_IO[3:0] obey the SPI_MOSI and SPI_MISO timing. In the 2-pin SPI Interface implementation, SPI_IO0 pin is the SPI Master-Out/Slave-In (MOSI) pin and the SPI_IO1 pin is the Master-In/Slave-out (MISO) pin.

TABLE 51-27: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			2	ns
T2	Data IN Setup Time	5.5			ns
T3	Data IN Hold Time	0			ns
Note:	Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow				