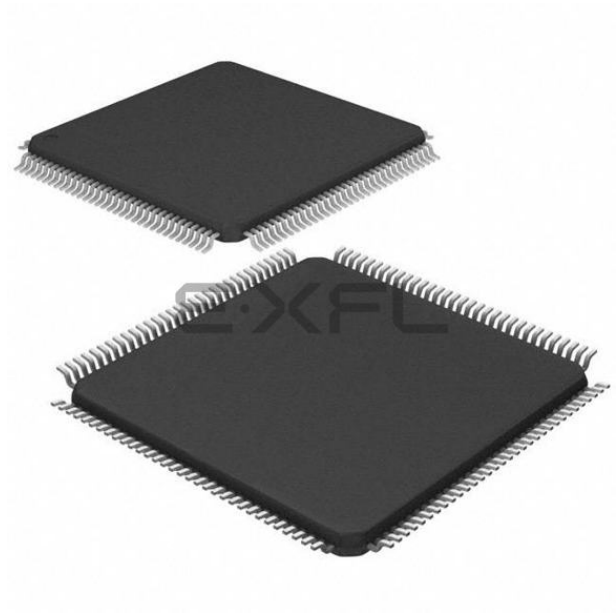


Welcome to E-XFL.COM

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	192KB
Interface	I²C, LPC, SMBus, SPI, UART
Number of I/O	106
Voltage - Supply	1.71V ~ 3.465V
Operating Temperature	-40°C ~ 85°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/mec1418-i-nu

TABLE 4-21: BAR INHIBIT DEVICE MAP

Bar Inhibit Bit	Logical Device Number
0	0h
1	1h
. . .	. . .
31	31h

4.11.6 LPC BAR INIT REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	BAR_Init This field is loaded into the LPC BAR at offset 60h on nSIO_RESET.	R/W	002Eh	nSIO_RESET

4.11.7 DEVICE MEMORY BAR INHIBIT REGISTER

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
63:0	Device Mem BAR_Inhibit[63:0] When bit i of the Device Mem BAR_Inhibit[63:0] field is asserted (‘1’), where i is the logical device number of one of the Device Memory Base Address Registers, the BAR for the associated device is disabled and its LPC Memory addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR. When bit i is not asserted (default), BAR activity for the Logical Device is based on the Valid bit in the BAR. All of the Device Mem BAR_Inhibit[63:0] bits are R/W and have no affect on reserved logical device numbers.	R/W	0h	nSYSRST

6.11.6 QMSPI BUFFER COUNT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_COUNT This is a count of the number of bytes currently valid in the Receive Buffer.	R	0h	RESET
15:0	TRANSMIT_BUFFER_COUNT This is a count of the number of bytes currently valid in the Transmit Buffer.	R	0h	RESET

6.11.7 QMSPI INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	R	-	-
14	RECEIVE_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
13	RECEIVE_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	1h	RESET
12	RECEIVE_BUFFER_FULL_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
11	Reserved	R	-	-
10	TRANSMIT_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
9	TRANSMIT_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	0h	RESET

TABLE 10-2: INTERRUPT SOURCE, ENABLE SET, ENABLE CLEAR, AND RESULT BIT ASSIGNMENTS (CONTINUED)

Aggregator IRQ	Aggregator Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description
GIRQ17	8	PECI Interface	PECIHOST	No	PECI Host Event
GIRQ17	9	TACH 0	TACH	No	Tachometer 0 Interrupt Event
GIRQ17	10	TACH 1	TACH	No	Tachometer 1 Interrupt Event
GIRQ18	0	Quad Master SPI Controller	QMSPI_INT	No	Master SPI Controller Requires Servicing
GIRQ19	0	eSPI_Slave	INTR_PC	No	Peripheral Channel Interrupt
GIRQ19	1	eSPI_Slave	INTR_BM1	No	Bus Mastering Channel 1 Interrupt
GIRQ19	2	eSPI_Slave	INTR_BM2	No	Bus Mastering Channel 2 Interrupt
GIRQ19	3	eSPI_Slave	INTR_LTR	No	Peripheral Message (LTR) Interrupt
GIRQ19	4	eSPI_Slave	INTR_OOB_UP	No	Out of Band Channel Up Interrupt
GIRQ19	5	eSPI_Slave	INTR_OOB_DOWN	No	Out of Band Channel Down Interrupt
GIRQ19	6	eSPI_Slave	INTR_FLASH	No	Flash Channel Interrupt
GIRQ19	7	eSPI_Slave	eSPI_RESET	No	eSPI_RESET
GIRQ19	8	MCHP Reserved	MCHP Reserved	-	-
GIRQ20	0	BC-Link 0 Master	BCM_BUSY_CLR	No	BC-Link Busy Clear Flag
GIRQ20	1	BC-Link 0 Master	BCM_ERR	No	BC-Link Error Flag Interrupt
GIRQ20	2	BC-Link 0 Master	BCM_INT	Yes	BC-Link Companion Interrupt Event
GIRQ20	3	BC-Link 1 Master	BCM_BUSY_CLR	No	BC-Link Busy Clear Flag
GIRQ20	4	BC-Link 1 Master	BCM_ERR	No	BC-Link Error Flag Interrupt
GIRQ20	5	BC-Link 1 Master	BCM_INT	Yes	BC-Link Companion Interrupt Event
GIRQ21	0-2	Test	Test	-	-
GIRQ22	0	LPC Interface	LPC_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - LPC Traffic Detected
GIRQ22	1	SMBus Controller 0	SMB_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMBus.0 START Detected
GIRQ22	2	SMBus Controller 1	SMB_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMBus.1 START Detected
GIRQ22	3	SMBus Controller 2	SMB_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMBus.2 START Detected
GIRQ22	4	PS2 Device Interface 0	PS2_DAT0_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - PS/2.0 Start Bit Detected
GIRQ22	5	PS2 Device Interface 1A	PS2_DAT1A_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - PS/2.1A Start Bit Detected

TABLE 10-4: JTVIC REGISTER SUMMARY (CONTINUED)

Offset	Register Name
21Ch	GIRQ15 Aggregator Control Register
220h	GIRQ16 Aggregator Control Register
224h	GIRQ17Aggregator Control Register
228h	GIRQ18 Aggregator Control Register
22Ch	GIRQ19 Aggregator Control Register
230h	GIRQ20 Aggregator Control Register
234h	GIRQ21 Aggregator Control Register
238h	GIRQ22 Aggregator Control Register
23Ch	GIRQ23 Aggregator Control Register
240h	GIRQ24 Aggregator Control Register
244h	GIRQ25 Aggregator Control Register
248h	GIRQ26 Aggregator Control Register
Interrupt Priority Control Registers	
300h	GIRQ8 [7:0] Interrupt Priority Register
304h	GIRQ8 [15:8] Interrupt Priority Register
308h	GIRQ8 [23:16] Interrupt Priority Register
30Ch	GIRQ8 [31:24] Interrupt Priority Register
310h	GIRQ9 [7:0] Interrupt Priority Register
314h	GIRQ9 [15:8] Interrupt Priority Register
318h	GIRQ9 [23:16] Interrupt Priority Register
31Ch	GIRQ9 [31:24] Interrupt Priority Register
320h	GIRQ10 [7:0] Interrupt Priority Register
324h	GIRQ10 [15:8] Interrupt Priority Register
328h	GIRQ10 [23:16] Interrupt Priority Register
32Ch	GIRQ10 [31:24] Interrupt Priority Register
330h	GIRQ11 [7:0] Interrupt Priority Register
334h	GIRQ11 [15:8] Interrupt Priority Register
338h	GIRQ11 [23:16] Interrupt Priority Register
33Ch	GIRQ11 [31:24] Interrupt Priority Register
340h	GIRQ12 [7:0] Interrupt Priority Register
344h	GIRQ12 [15:8] Interrupt Priority Register
348h	GIRQ12 [23:16] Interrupt Priority Register

12.9.7 EC DATA BYTE 2 REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_2 This is byte 2 of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 12.8.2, "EC Data Register".	R/W	0h	nSYSRST

12.9.8 EC DATA BYTE 3 REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_3 This is byte 3 (Most Significant Byte) of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 12.8.2, "EC Data Register".	R/W	0h	nSYSRST

12.9.9 INTERRUPT SOURCE LSB REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_LSB EC Software Interrupt Least Significant Bits. These bits are software interrupt bits that may be set by the EC to notify the host of an event. The meaning of these bits is dependent on the firmware implementation. Each bit in this field is cleared when written with a '1b'. The ability to clear the bit can be disabled by the EC if the corresponding bit in the Host Clear Enable Register is set to '0b'. This may be used by firmware for events that cannot be cleared while the event is still active.	R/WC	0h	nSYSRST
0	EC_WR EC Mailbox Write. This bit is set when the EC-to-HOST Mailbox Register has been written by the EC at offset 01h of the EC-Only registers.	R	0h	nSYSRST

14.13.10 EC BYTE CONTROL REGISTER

This register is aliased to the OS Byte Control Register on page 233. The OS Byte Control Register is a read only version of this register.

Offset	105h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	R	-	-
0	FOUR_BYTE_ACCESS See FOUR_BYTE_ACCESS (see Note) bit in OS Byte Control Register on page 233 for bit description.	R/W	0b	nSYSRST

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 16-4, "GATEA20 Command/Data Sequence Examples"). The foregoing description assumes that the SAEN configuration bit is reset.

When the MEC140x/1x 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 EC Keyboard Status Register be activated; for example, this host control of GATEA20 is transparent to firmware, with no consequent degradation of overall system performance. TABLE 16-4: details the possible GATEA20 sequences and the MEC140x/1x responses.

An additional level of control flexibility is offered via a memory-mapped synchronous set and reset capability. Any data written to the SETGA20L Register causes the GATEA20 host latch to be set; any data written to the RSTGA20L Register 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 SETGA20L and RSTGA20L registers, firmware should read back the GATEA20 status via the GATEA20 Control Register (with SAEN = 0) to confirm the actual GATEA20 response.

TABLE 16-4: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES

Command(C) / Data (D)	R/W	D[7:0]	IBF Flag	GATEA20	Comments
C D C	W W W	D1 DF FF	0 0 0	Q 1 1	GATEA20 Turn-on Sequence
C D C	W W W	D1 DD FF	0 0 0	Q 0 0	GATEA20 Turn-off Sequence
C C D C	W W W W	D1 D1 DF FF	0 0 0 0	Q Q 1 1	GATEA20 Turn-on Sequence(*)
C C D C	W W W W	D1 D1 DD FF	0 0 0 0	Q Q 0 0	GATEA20 Turn-off Sequence(*)
C C C	W W W	D1 XX** FF	0 1 1	Q Q Q	Invalid Sequence

17.6 Power, Clocks and Reset

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

17.6.1 POWER DOMAINS

Name	Description
VTR	This Power Well is used to power the registers and logic in this block.

17.6.2 CLOCK INPUTS

Name	Description
1.8432MHz_Clk	The UART requires a 1.8432 MHz ± 2% clock input for baud rate generation.
24MHz_Clk	24 MHz ± 2% clock input. This clock may be enabled to generate the baud rate, which requires a 1.8432 MHz ± 2% clock input.

17.6.3 RESETS

Name	Description
nSYSRST	This reset is asserted when VTR is applied.
nSIO_RESET	This is an alternate reset condition, typically asserted when the main power rail is asserted.
RESET	This reset is determined by the POWER bit signal. When the power bit signal is 1, this signal is equal to nSIO_RESET. When the power bit signal is 0, this signal is equal to nSYSRST.

17.7 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 17-8, “Interrupt Control,” on page 278.

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 17-8, “Interrupt Control,” on page 278.

TABLE 21-10: SPISR ENCODING

SPISR Value	Sub-Second Interrupt Rate, Hz	Interrupt Period
0	Interrupts disabled	
1	2	500 ms
2	4	250 ms
3	8	125 ms
4	16	62.5 ms
5	32	31.25 ms
6	64	15.63 ms
7	128	7.813 ms
8	256	3.906 ms
9	512	1.953 ms
10	1024	977 µS
11	2048	488 µS
12	4096	244 µS
13	8192	122 µS
14	16384	61 µS
15	32768	30.5 µS

21.10.6 SUB-WEEK CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	R	-	-
9:7	SUBWEEK_TICK This field selects the clock source for the Sub-Week Counter. See Table 21-7, "Sub-Week Alarm Counter Clock" for the description of the options for this field. See also Note 1.	R/W	0	VBAT_POR
6	AUTO_RELOAD 1= No reload occurs when the Sub-Week Counter expires 0= Reloads the SUBWEEK_COUNTER_LOAD field into the Sub-Week Counter when the counter expires.	R/W	0	VBAT_POR
5	SYSPWR_PRES_ENABLE Enables SYSPWR_PRES Pin to disable Week the Week timer and Sub-Week Timer Power-Up Events from driving VCI_OUT high 1=The SYSPWR_PRES Pin input low disables both the Week timer and Sub-Week Timer Power-Up Events from driving VCI_OUT high 0=The SYSPWR_PRES Pin input has no effect on the Week timer and Sub-Week Timer Power-Up Events driving VCI_OUT high	R/W	0	VBAT_POR

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.

24.11.4 DMA CHANNEL N DEVICE ADDRESS REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DEVICE_ADDRESS This is the Master Device address. This is used as the address that will access the Device on the DMA. The Device is defined as the Master of the DMA transfer; as in the device that is controlling the Hardware Flow Control. **APPLICATION NOTE:** Only Channel 0 has CRC function which may be utilized only by the Quad SPI Master Controller and for Memory-to-Memory transfers. It is recommended to use Channels 1-6 for the SMBus Controllers. This field is updated by Hardware after every Data Packet transfer by the size of the transfer, as defined by DMA Channel Control:Transfer Size while the DMA Channel Control:Increment Device Address is Enabled. **Note:** This field is only as large as the maximum allowed AHB Address Size in the system. If the HADDR size is 24 Bits, then Bits [31:24] will be RESERVED.	R/W	0000h	DMA_RESET

24.11.5 DMA CHANNEL N CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	R	-	-
25	TRANSFER_ABORT This is used to abort the current transfer on this DMA Channel. The aborted transfer will be forced to terminate immediately.	R/W	0h	DMA_RESET
24	TRANSFER_GO This is used for the **Firmware Flow Control** DMA transfer. This is used to start a transfer under the **Firmware Flow Control**. Do not use this in conjunction with the **Hardware Flow Control**; **DMA Channel Control:Disable Hardware Flow Control** must be set in order for this field to function correctly.	R/W	0h	DMA_RESET
23	Reserved	R	-	-
22:20	TRANSFER_SIZE This is the transfer size in Bytes of each Data Packet transfer. **Note:** The transfer size must be a legal transfer size. Valid sizes are 1, 2 and 4 Bytes.	R/W	0h	DMA_RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
19	DISABLE_HARDWARE_FLOW_CONTROL This will Disable the **Hardware Flow Control**. When disabled, any DMA Master device attempting to communicate to the DMA over the **DMA Flow Control Interface (Ports: dma_req, dma_term, and dma_done)** will be ignored. This should be set before using the DMA channel in **Firmware Flow Control** mode.	RW	0h	DMA_RESET
18	LOCK_CHANNEL This is used to lock the arbitration of the Channel Arbiter on this channel once this channel is granted. Once this is locked, it will remain on the arbiter until it has completed it transfer (either the Transfer Aborted, Transfer Done or Transfer Terminated conditions). **Note:** This setting may starve other channels if the locked channel takes an excessive period of time to complete.	RW	0h	DMA_RESET
17	INCREMENT_DEVICE_ADDRESS This will enable an auto-increment to the **DMA Channel Device Address.** **1:** Increment the **DMA Channel Device Address** by **DMA Channel Control:Transfer Size** after every **Data Packet transfer** **0:** Do nothing	RW	0h	DMA_RESET
16	INCREMENT_MEMORY_ADDRESS This will enable an auto-increment to the **DMA Channel Memory Address.** 1=Increment the DMA Channel Memory Address by DMA Channel Control:Transfer Size after every Data Packet transfer 0=Do nothing **Note:** *If this is not set, the DMA will never terminate the transfer on its own. It will have to be terminated through the* Hardware Flow Control *or through a* DMA Channel Control:Transfer Abort.	RW	0h	DMA_RESET
15:9	HARDWARE_FLOW_CONTROL_DEVICE This is the device that is connected to this channel as its **Hardware Flow Control** master. The **Flow Control Interface** is a bus with each master concatenated onto it. This selects which bus index of the concatenated **Flow Control Interface** bus is targeted towards this channel. The **Flow Control Interface** Port list is **dma_req, dma_term,** and **dma_done**.	RW	0h	DMA_RESET
8	TRANSFER_DIRECTION This determines the direction of the DMA Transfer. 1=Data Packet Read from **Memory Start Address** followed by Data Packet Write to **Device Address** 0=Data Packet Read from **Device Address** followed by Data Packet Write to **Memory Start Address**	RW	0h	DMA_RESET
7:6	Reserved	R	-	-

25.10 Instance Description

There is one instance of the PECI Core implemented in the PECI Interface in the MEC140x/1x. See PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, SMSC Confidential, 4/15/11 for a description of the PECI Core.

25.11 PECI Interface Registers

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

TABLE 25-3: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
PECI Interface	0	EC	32-bit Internal Address Space	0000_6400h

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

TABLE 25-4: PECI INTERFACE REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	Write Data Register
04h	Read Data Register
08h	Control Register
0Ch	Status Register 1
10h	Status Register 2
14h	Error Register
18h	Interrupt Enable 1 Register
1Ch	Interrupt Enable 2 Register
20h	Optimal Bit Time Register (Low Byte)
24h	Optimal Bit Time Register (High Byte)
28h	Test
2Ch	Test
30h-3Ch	Reserved
40h	Block ID Register
44h	Revision Register
48h - 7Ch	Test

Note: Test registers are reserved for Microchip use only. Reading and writing Test registers may cause undesirable results

For register details see References [1].

TABLE 27-3: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	PWMx Counter ON Time Register
04h	PWMx Counter OFF Time Register
08h	PWMx Configuration Register

27.11.1 PWMX COUNTER ON TIME REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	PWMX_COUNTER_ON_TIME This field determine both the frequency and duty cycle of the PWM signal. When this field is set to zero and the PWMX_COUNTER_OFF_-TIME is not set to zero, the PWM_OUTPUT is held low (Full Off).	R/W	0000h	nSYSRST

27.11.2 PWMX COUNTER OFF TIME REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	R	-	-
15:0	PWMX_COUNTER_OFF_TIME This field determine both the frequency and duty cycle of the PWM signal. When this field is set to zero, the PWM_OUTPUT is held high (Full On).	R/W	FFFFh	nSYSRST

27.11.3 PWMX CONFIGURATION REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	R	-	-
6:3	CLOCK_PRE_DIVIDER The Clock source for the 16-bit down counter (see PWMx Counter ON Time Register and PWMx Counter OFF Time Register) 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.	R/W	0000b	nSYSRST

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	UPDATE_INTERVAL1 The number of PWM periods between updates to current duty cycle when the segment index is equal to 001b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST
3:0	UPDATE_INTERVAL0 The number of PWM periods between updates to current duty cycle when the segment index is equal to 000b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	nSYSRST

TABLE 31-5: FREQUENCY SETTINGS

Divider	Frequency
1	24MHz
2	16MHz
3	12MHz
4	9.6MHz
15	2.18MHz
2A	1.12MHz

35.0 VBAT REGISTER BANK

35.1 Introduction

This chapter defines a bank of registers powered by VBAT.

35.2 Interface

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

35.3 Power, Clocks and Reset

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

35.3.1 POWER DOMAINS

Name	Description
VBAT	The VBAT Register Bank are all implemented on this single power domain.

35.3.2 CLOCK INPUTS

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

35.3.3 RESETS

Name	Description
VBAT_POR	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

35.4 Interrupts

Name	Description
PFR_Status	This interrupt signal from the Power-Fail and Reset Status Register indicates VBAT RST and WDT events.

35.5 Low Power Modes

The VBAT Register Bank is designed to always operate in the lowest power consumption state.

35.6 Description

The VBAT Register Bank block is a block implemented for aggregating miscellaneous battery-backed registers required the host and by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

36.7 Power, Clocks and Reset

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

36.7.1 POWER DOMAINS

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

36.7.2 CLOCK INPUTS

No special clocks are required for this block.

36.7.3 RESETS

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

36.8 Interrupts

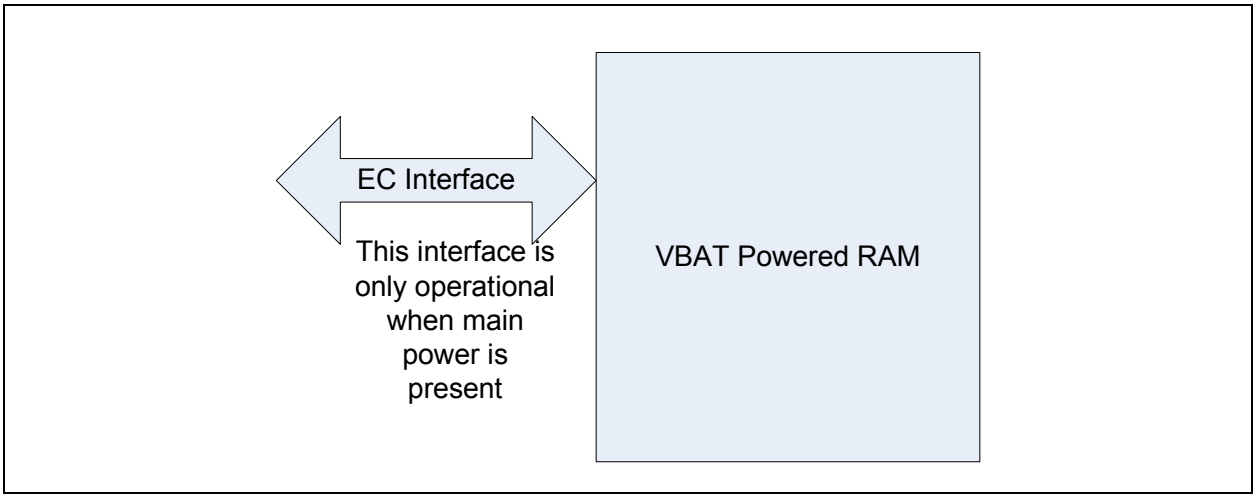
This block does not generate any interrupts.

36.9 Low Power Modes

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

36.10 Description

FIGURE 36-2: VBAT RAM BLOCK DIAGRAM



43.16 Keyboard Scan Matrix Timing

TABLE 43-16: ACTIVE PRE DRIVE MODE TIMING

Parameter	Symbol	Value			Units	Notes
		MIN	TYP	MAX		
Active Predrive Mode	$t_{PREDRIVE}$	40.87	41.7	42.5	ns	

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)
8111C	GPIO	0	GPIO Registers	GPIO107 Pin Control	4
81120	GPIO	0	GPIO Registers	GPIO110 Pin Control	4
81124	GPIO	0	GPIO Registers	GPIO111 Pin Control	4
81128	GPIO	0	GPIO Registers	GPIO112 Pin Control	4
8112C	GPIO	0	GPIO Registers	GPIO113 Pin Control	4
81130	GPIO	0	GPIO Registers	GPIO114 Pin Control	4
81134	GPIO	0	GPIO Registers	GPIO115 Pin Control	4
81138	GPIO	0	GPIO Registers	GPIO116 Pin Control	4
8113C	GPIO	0	GPIO Registers	GPIO117 Pin Control	4
81140	GPIO	0	GPIO Registers	GPIO120 Pin Control	4
81144	GPIO	0	GPIO Registers	GPIO121 Pin Control	4
81148	GPIO	0	GPIO Registers	GPIO122 Pin Control	4
8114C	GPIO	0	GPIO Registers	GPIO123 Pin Control	4
81150	GPIO	0	GPIO Registers	GPIO124 Pin Control	4
81154	GPIO	0	GPIO Registers	GPIO125 Pin Control	4
81158	GPIO	0	GPIO Registers	GPIO126 Pin Control	4
8115C	GPIO	0	GPIO Registers	GPIO127 Pin Control	4
81160	GPIO	0	GPIO Registers	GPIO130 Pin Control	4
81164	GPIO	0	GPIO Registers	GPIO131 Pin Control	4
81168	GPIO	0	GPIO Registers	GPIO132 Pin Control	4
8116C	GPIO	0	GPIO Registers	GPIO133 Pin Control	4
81170	GPIO	0	GPIO Registers	GPIO134 Pin Control	4
81174	GPIO	0	GPIO Registers	GPIO135 Pin Control	4
81178	GPIO	0	GPIO Registers	GPIO136 Pin Control	4
81180	GPIO	0	GPIO Registers	GPIO140 Pin Control	4
81184	GPIO	0	GPIO Registers	GPIO141 Pin Control	4
81188	GPIO	0	GPIO Registers	GPIO142 Pin Control	4
8118C	GPIO	0	GPIO Registers	GPIO143 Pin Control	4
81190	GPIO	0	GPIO Registers	GPIO144 Pin Control	4
81194	GPIO	0	GPIO Registers	GPIO145 Pin Control	4
81198	GPIO	0	GPIO Registers	GPIO146 Pin Control	4
8119C	GPIO	0	GPIO Registers	GPIO147 Pin Control	4
811A0	GPIO	0	GPIO Registers	GPIO150 Pin Control	4
811A4	GPIO	0	GPIO Registers	GPIO151 Pin Control	4
811A8	GPIO	0	GPIO Registers	GPIO152 Pin Control	4
811AC	GPIO	0	GPIO Registers	GPIO153 Pin Control	4
811B0	GPIO	0	GPIO Registers	GPIO154 Pin Control	4
811B4	GPIO	0	GPIO Registers	GPIO155 Pin Control	4
811B8	GPIO	0	GPIO Registers	GPIO156 Pin Control	4