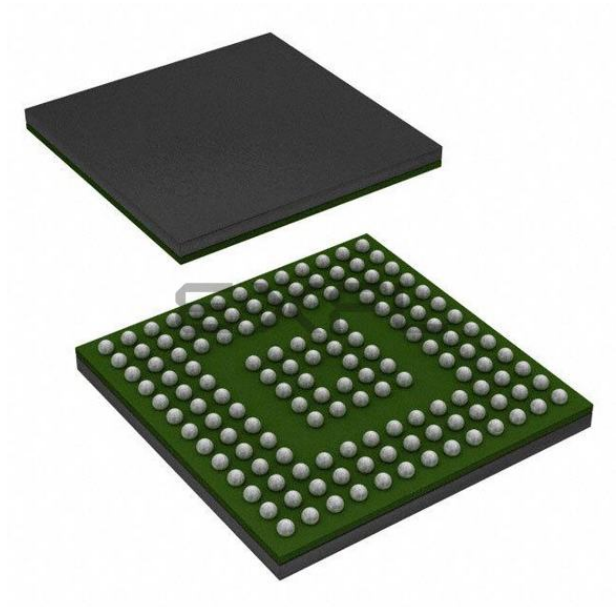


Welcome to E-XFL.COM

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	0°C ~ 70°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/mec1701q-c2-sz

2.0 PIN CONFIGURATION

2.1 Description

The Pin Configuration chapter includes Pin List By Pin Name, Signal Description by Signal, Notes for Tables in this Chapter, Pin Default State Through Power Transitions, and Packages.

2.2 Terminology and Symbols for Pins/Buffers

2.2.1 BUFFER TERMINOLOGY

Term	Definition
#	The '#' sign at the end of a signal name indicates an active-low signal
n	The lowercase 'n' preceding a signal name indicates an active-low signal
PWR	Power
PIO	Programmable as Input, Output, Open Drain Output, Bi-directional or Bi-directional with Open Drain Output. Configurable drive strength from 2ma to12ma. **Note:** All GPIOs have programmable drive strength options of 2ma, 4ma, 8ma and 12ma. GPIO pin drive strength is determined by the DRIVE_STRENGTH field in the Pin Control 2 Register.
In	I Type Input Buffer.
O2ma	O-2 mA Type Buffer.
PCI	PCI pin. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 1)
PECI	PECI Input/Output. These pins operate at the processor voltage level (VREF_VTT)
SB-TSI	SB-TSI Input/Output. These pins operate at the processor voltage level (VREF_VTT)

Note 1: See the "PCI Local Bus Specification," Revision 2.1, Section 4.2.2.

2: See the "PCI Local Bus Specification," Revision 2.1, Section 4.2.2 and 4.2.3.

2.2.2 PIN NAMING CONVENTIONS

- Pin Name is composed of the multiplexed options separated by '/'. E.g., GPIOxxxx/SignalA/SignalB.
- Parenthesis '()' are used to list aliases or alternate functionality for a single mux option. For example, GPIO062/(RESETO#) has only a single mux option, GPIO062, but the signal GPIO062 can also be used or interpreted as RESETO#.
- Signal Names appended with a numeric value indicates the Instance Number E.g., PWM0, PWM1, etc. indicates that PWM0 is the PWM output for PWM Instance 0, PWM1 is the PWM output for PWM Instance 1, etc. Note that this same instance number is shown in the Register Base Address tables linking the specific PWM block instance to a specific signal on the pinout. The instance number may be omitted if there in only one instance of the IP block implemented.

2.3 Notes for Tables in this Chapter

Note	Description
Note 1	The nEC_SCI pin can be controlled by hardware and EC firmware. The nEC_SCI pin can drive either the ACPI Run-time GPE Chipset input or the Wake GPE Chipset input. Depending how the nEC_SCI pin is used, other ACPI-related SCI functions may be best supplied by other general purpose outputs that can be configured as open-drain drivers.
Note 2	These pins require an external weak pull-up resistors of 10k-100k ohms.
Note 3	A weak pull-up resistor is recommended on the BC-Link data line (100KΩ).
Note 4	The UARTs can be used by the Host or EC. This pin can be VCC protected or not VCC protected under program control by the POWER bit in the Configuration Select Register in Host configuration space (also accessible by the EC).

MEC1703-169 WFBGA-XY	Signal	Power Rail	Pad Type	Glitch Prot	Over-voltage Prot	Under-voltage Prot	Backdrive Prot
P9	GPIO130/I2C10_SDA/TOUT1	VTR2	PIO	X		X	X
P10	GPIO021/LPCPD#/KSI2	VTR2	PIO	X		X	X
P11	GPIO014/PWM6/GPTP-IN6	VTR2	PIO	X		X	X
P12	GPIO026/TIN1/KSI3	VTR2	PIO	X		X	X
P13	GPIO115/PS2_DAT0A	VTR2	PIO	X		X	X
P14	GPIO053/PWM0/GPWM0	VTR2	PIO	X	X	X	X
P15	GPIO027/TIN2/KSI4	VTR2	PIO	X		X	X
R2	GPIO065/PCI_CLK/ESPI_CLK	VTR3	PIO	X		X	X
R3	GPIO070/LAD0/ESPI_IO0	VTR3					
	GPIO070/ESPI_IO0		PIO				
	LAD0		PCI				
R4	GPIO071/LAD1/ESPI_IO1	VTR3					
	GPIO071/ESPI_IO1		PIO				
	LAD1		PCI				
R5	GPIO073/LAD3/ESPI_IO3	VTR3					
	GPIO073/ESPI_IO3		PIO				
	LAD3		PCI				
R6	GPIO224/GPTP-IN4/SHD_IO1	VTR2	PIO	X		X	X
R7	GPIO223/SHD_IO0	VTR2	PIO	X		X	X
R8	GPIO013/I2C07_SCL/TOUT2	VTR2	PIO	X		X	X
R9	GPIO020/KSI1	VTR2	PIO	X		X	X
R10	GPIO131/I2C10_SCL/TOUT0	VTR2	PIO	X		X	X
R11	GPIO230	VTR2	PIO	X		X	X
R12	GPIO001/PWM4	VTR2	PIO	X		X	X
R13	GPIO152/GPTP-OUT3/KSO16	VTR2	PIO	X		X	X
R14	GPIO132/I2C06_SDA/KSO14	VTR2	PIO	X		X	X

2.6 Signal Description by Signal

EMULATED POWER WELL

Power well emulation for GPIOs and for signals that are multiplexed with GPIO signals is controlled by the POWER_-GATING field in the GPIO Pin Control Register. Power well emulation for signals that are not multiplexed with GPIO signals is defined by the entries in this column.

GATED STATE

This column defines the internal value of an input signal when either its emulated power well is inactive or it is not selected by the GPIO alternate function MUX. A value of “No Gate” means that the internal signal always follows the pin even when the emulated power well is inactive.

Signal	Emulated Power Rail	Gated State	Notes
GPIO047	VTR	No Gate	
GPIO050	VTR	No Gate	
GPIO051	VTR	No Gate	
GPIO052	VTR	No Gate	
GPIO053	VTR	No Gate	
GPIO054	VTR	No Gate	
GPIO055	VTR	No Gate	
GPIO056	VTR	No Gate	
GPIO057	VTR	No Gate	
GPIO060	VTR	No Gate	
GPIO061	VTR	No Gate	
GPIO062	VTR	No Gate	Note 7
GPIO063	VTR	No Gate	
GPIO064	VTR	No Gate	
GPIO065	VTR	No Gate	
GPIO066	VTR	No Gate	
GPIO067	VTR	No Gate	
GPIO070	VTR	No Gate	
GPIO071	VTR	No Gate	
GPIO072	VTR	No Gate	
GPIO073	VTR	No Gate	
GPIO100	VTR	No Gate	
GPIO101	VTR	No Gate	
GPIO102	VTR	No Gate	
GPIO104	VTR	No Gate	
GPIO105	VTR	No Gate	
GPIO106	VTR	No Gate	
GPIO107	VTR	No Gate	
GPIO110	VTR	No Gate	
GPIO111	VTR	No Gate	
GPIO112	VTR	No Gate	
GPIO113	VTR	No Gate	
GPIO114	VTR	No Gate	
GPIO115	VTR	No Gate	
GPIO120	VTR	No Gate	
GPIO121	VTR	No Gate	
GPIO122	VTR	No Gate	
GPIO123	VTR	No Gate	
GPIO124	VTR	No Gate	
GPIO125	VTR	No Gate	
GPIO126	VTR	No Gate	
GPIO127	VTR	No Gate	
GPIO130	VTR	No Gate	

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
SPI Controllers									
D13	D11	D11	D11	B12	C12	D14	PVT_CLK	Quad SPI Controller Clock, Private SPI port	
C12	C11	C11	C11	B13	E11	E12	PVT_CS#	Quad SPI Controller Chip Select, Private SPI port	
E12	F12	F12	F12	D13	F10	F12	PVT_IO0	Quad SPI Controller Data 0, Private SPI port	
E13	E11	E11	E11	D12	D12	F14	PVT_IO1	Quad SPI Controller Data 1, Private SPI port	
D12	E12	E12	E12	C13	F9	G12	PVT_IO2	Quad SPI Controller Data 2, Private SPI port	
C13	D12	D12	D12	C12	C13	E14	PVT_IO3	Quad SPI Controller Data 3, Private SPI port	
N7	N5	N5	N5	N9	J7	M9	SHD_CLK	Quad SPI Controller Clock, Shared SPI port	
N8	N4	N4	N4	M8	L7	M8	SHD_CS#	Quad SPI Controller Chip Select, Shared SPI port	
N5	M6	M6	M6	M7	K7	R7	SHD_IO0	Quad SPI Controller Data 0, Shared SPI port	
M7	M5	M5	M5	N6	M3	R6	SHD_IO1	Quad SPI Controller Data 1, Shared SPI port	
M6	L6	L6	L6	N7	M4	P7	SHD_IO2	Quad SPI Controller Data 2, Shared SPI port	
N6	N3	N3	N3	N8	M5	M7	SHD_IO3	Quad SPI Controller Data 3, Shared SPI port	
	H12	H12	H12	E13	H12	H15	SPI0_CLK	GP-SPI SPI Clock	
B5	F5	F5	F5	C6	D7	B7	SPI0_CS#	GP-SPI Chip Select	
	H13	H13	H13	F12	H13	J12	SPI0_MISO	GP-SPI SPI Output	
D7	C6	C6	C6	B5	B6	D7	SPI0_MOSI	GP-SPI SPI Input	
A6	B7	B7	B7	A7	A7	B8	SPI1_CLK	GP-SPI SPI Clock	
D8	E7	E7	E7	D4	B5	E8	SPI1_CS#	GP-SPI Chip Select	
B6	F7	F7	F7	F7	A5	D8	SPI1_MOSI	GP-SPI SPI Output	
A5	A7	A7	A7	A6	B7	A8	SPI1_MISO	GP-SPI SPI Input	
VBAT-Powered Control Interface									
A3	A5	A5	A5	B4	C6	D6	BGPO0	VBAT driven GPO	
	A4	A4	A4	C5	E6	F6	BGPO1	VBAT driven GPO	Note 8
	B5	B5	B5	B3	C5	A5	BGPO2	VBAT driven GPO	Note 8
	B6	B6	B6	F5	D6	E6	BGPO3	VBAT driven GPO	Note 8
				A2	C3	B5	BGPO4	VBAT driven GPO	Note 8
				E4	A3	B3	BGPO5	VBAT driven GPO	Note 8
D6	E6	E6	E6	F6	C7	B6	VCI_IN0#	Input can cause wakeup or interrupt event, active low	Note 13
D5	A6	A6	A6	A4	C4	A6	VCI_IN1#	Input can cause wakeup or interrupt event, active low	Note 13
B3	C4	C4	C4	C3	D5	E5	VCI_IN2#	Input can cause wakeup or interrupt event, active low	Note 13

Legend	Notes
(P) = I/O state is driven by protocol while power is applied. **Z** = Tristate **In** = Input	Note A: Pin exhibits "VCC" power domain emulation. Note B: Pin is programmable by the EC and retains its value through a VTR power cycle. Note C: Pin is programmable by the EC and affected by other VBAT inputs pins. Note D: Pin exhibits "VTR" power domain emulation. Note E: Does not include GPIO042, GPIO043, and GPIO062

TABLE 2-4: PIN DEFAULT STATE THROUGH POWER TRANSITIONS

Signal	VBAT Applied	VBAT Stable	VTR Applied	nSYS _ RST De-asserted	VCC_ PWRGD Asserted	VCC_ PWRGD De-asserted	RESET_ SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
nSMI	N/A	N/A	N/A	N/A	1> OD(P)> 1	OD(1)	In	glitch	N/A	
KBRST	N/A	N/A	N/A	N/A	1> OD(P)> 1	Z	Z>In	glitch	N/A	Note F
A20M	N/A	N/A	N/A	N/A	1> OD(P)> 1	Z	Z	glitch	N/A	Note F
LPCPD#	N/A	N/A	N/A	N/A	In	Z	Z	glitch	N/A	Note F

Legend	Notes
(P) = I/O state is driven by protocol while power is applied. **Z** = Tristate **In** = Input **OD** = Open Drain Output Undriven (1) or driven (0)	Note F: Pin is programmable by the EC and retains its value through a VTR power cycle

Agg IRQ	Agg Bit	HWB Instance Name	Interrupt Event	Wake Event	Source Description	Agg NVIC	Direct NVIC
GIRQ16	0	Public Key Engine	PKE ERROR	No	PKE core error detected	8	65
	1	Public Key Engine	PKE END	No	PKE completed processing		66
	2	Random Number Generator	RNG	No	RNG completed processing		67
	3	AES	AES	No	Interrupt from AES block		68
	4	Hash	HASH	No	Interrupt from SHA block		69
	5-31	Reserved					
GIRQ17	0	PECI	PECI_INT	No	PECI Host Event	9	70
	1	TACH 0	TACH	No	Tachometer 0 Interrupt Event		71
	2	TACH 1	TACH	No	Tachometer 1 Interrupt Event		72
	3	TACH 2	TACH	No	Tachometer 2 Interrupt Event		73
	4	RPM2PWM 0	FAN_FAIL	No	Failure to achieve target RPM		74
	5	RPM2PWM 0	FAN_STALL	No	Fan stall condition		75
	6	RPM2PWM 1	FAN_FAIL	No	Failure to achieve target RPM		76
	7	RPM2PWM 1	FAN_STALL	No	Fan stall condition		77
	8	ADC Controller	ADC_Single_Int	No	ADC Controller - Single-Sample ADC Conversion Event		78
	9	ADC Controller	ADC_Repeat_Int	No	ADC Controller - Repeat-Sample ADC Conversion Event		79
	10	RC-ID 0	RCID	No	0-1 transition of RC-ID done flag		80
	11	RC-ID 1	RCID	No	0-1 transition of RC-ID done flag		81
	12	RC-ID 2	RCID	No	0-1 transition of RC-ID done flag		82
	13	Breathing LED 0	PWM_WDT	No	Blinking LED 0 Watchdog Event		83
	14	Breathing LED 1	PWM_WDT	No	Blinking LED 1 Watchdog Event		84
	15	Breathing LED 2	PWM_WDT	No	Blinking LED 2 Watchdog Event		85
	16	Breathing LED 3	PWM_WDT	No	Blinking LED 3 Watchdog Event		86
	17-24		Reserved				
	25	RTOS Timer	SWI_0	No	Soft Interrupt request 0		
	26	RTOS Timer	SWI_1	No	Soft Interrupt request 1		
	27	RTOS Timer	SWI_2	No	Soft Interrupt request 2		
	28	RTOS Timer	SWI_3	No	Soft Interrupt request 3		
	29-31	Reserved					

3.7 Register Map

Block	Instance	Register	Register Address
Watchdog Timer	0	WDT Load Register	40000000h
Watchdog Timer	0	WDT Control Register	40000004h
Watchdog Timer	0	WDT Kick Register	40000008h
Watchdog Timer	0	WDT Count Register	4000000Ch
16-bit Basic Timer	0	Timer Count Register	40000C00h
16-bit Basic Timer	0	Timer Preload Register	40000C04h
16-bit Basic Timer	0	Timer Status Register	40000C08h
16-bit Basic Timer	0	Timer Int Enable Register	40000C0Ch
16-bit Basic Timer	0	Timer Control Register	40000C10h
16-bit Basic Timer	1	Timer Count Register	40000C20h
16-bit Basic Timer	1	Timer Preload Register	40000C24h
16-bit Basic Timer	1	Timer Status Register	40000C28h
16-bit Basic Timer	1	Timer Int Enable Register	40000C2Ch
16-bit Basic Timer	1	Timer Control Register	40000C30h
16-bit Basic Timer	2	Timer Count Register	40000C40h
16-bit Basic Timer	2	Timer Preload Register	40000C44h
16-bit Basic Timer	2	Timer Status Register	40000C48h
16-bit Basic Timer	2	Timer Int Enable Register	40000C4Ch
16-bit Basic Timer	2	Timer Control Register	40000C50h
16-bit Basic Timer	3	Timer Count Register	40000C60h
16-bit Basic Timer	3	Timer Preload Register	40000C64h
16-bit Basic Timer	3	Timer Status Register	40000C68h
16-bit Basic Timer	3	Timer Int Enable Register	40000C6Ch
16-bit Basic Timer	3	Timer Control Register	40000C70h
32-bit Basic Timer	0	Timer Count Register	40000C80h
32-bit Basic Timer	0	Timer Preload Register	40000C84h
32-bit Basic Timer	0	Timer Status Register	40000C88h
32-bit Basic Timer	0	Timer Int Enable Register	40000C8Ch
32-bit Basic Timer	0	Timer Control Register	40000C90h
32-bit Basic Timer	1	Timer Count Register	40000CA0h
32-bit Basic Timer	1	Timer Preload Register	40000CA4h
32-bit Basic Timer	1	Timer Status Register	40000CA8h
32-bit Basic Timer	1	Timer Int Enable Register	40000CACh
32-bit Basic Timer	1	Timer Control Register	40000CB0h
16-bit Counter Timer	0	Timer x Control Register	40000D00h
16-bit Counter Timer	0	Timer x Clock and Event Control Register	40000D04h
16-bit Counter Timer	0	Timer x Reload Register	40000D08h
16-bit Counter Timer	0	Timer x Count Register	40000D0Ch
16-bit Counter Timer	1	Timer x Control Register	40000D20h
16-bit Counter Timer	1	Timer x Clock and Event Control Register	40000D24h
16-bit Counter Timer	1	Timer x Reload Register	40000D28h

Block	Instance	Register	Register Address
DMA Channel	11	DMA Channel N Device Address	4000270Ch
DMA Channel	11	DMA Channel N Control Register	40002710h
DMA Channel	11	DMA Channel N Interrupt Status Register	40002714h
DMA Channel	11	DMA Channel N Interrupt Enable Register	40002718h
DMA Channel	11	TEST	4000271Ch
DMA Channel	12	DMA Channel N Activate Register	40002740h
DMA Channel	12	DMA Channel N Memory Start Address Register	40002744h
DMA Channel	12	DMA Channel N Memory End Address Register	40002748h
DMA Channel	12	DMA Channel N Device Address	4000274Ch
DMA Channel	12	DMA Channel N Control Register	40002750h
DMA Channel	12	DMA Channel N Interrupt Status Register	40002754h
DMA Channel	12	DMA Channel N Interrupt Enable Register	40002758h
DMA Channel	12	TEST	4000275Ch
DMA Channel	13	DMA Channel N Activate Register	40002780h
DMA Channel	13	DMA Channel N Memory Start Address Register	40002784h
DMA Channel	13	DMA Channel N Memory End Address Register	40002788h
DMA Channel	13	DMA Channel N Device Address	4000278Ch
DMA Channel	13	DMA Channel N Control Register	40002790h
DMA Channel	13	DMA Channel N Interrupt Status Register	40002794h
DMA Channel	13	DMA Channel N Interrupt Enable Register	40002798h
DMA Channel	13	TEST	4000279Ch
EEPROM Controller	0	EEPROM Mode Register	40002C00h
EEPROM Controller	0	EEPROM Execute Register	40002C04h
EEPROM Controller	0	EEPROM Status Register	40002C08h
EEPROM Controller	0	EEPROM Interrupt Enable Register	40002C0Ch
EEPROM Controller	0	EEPROM Password Register	40002C10h
EEPROM Controller	0	EEPROM Unlock Register	40002C14h
EEPROM Controller	0	EEPROM lock Register	40002C18h
EEPROM Controller	0	TEST	40002C1Ch
EEPROM Controller	0	EEPROM Buffer Register	40002C20h
SMB-I2C	0	Control Register	40004000h
SMB-I2C	0	Status Register	40004000h
SMB-I2C	0	Own Address Register	40004004h
SMB-I2C	0	Data Register	40004008h
SMB-I2C	0	Master Command Register	4000400Ch
SMB-I2C	0	Slave Command Register	40004010h
SMB-I2C	0	PEC Register	40004014h
SMB-I2C	0	Repeated START Hold Time Register	40004018h
SMB-I2C	0	Completion Register	40004020h
SMB-I2C	0	Idle Scaling Register	40004024h
SMB-I2C	0	Configuration Register	40004028h
SMB-I2C	0	Bus Clock Register	4000402Ch
SMB-I2C	0	Block ID Register	40004030h

The controller will accesses SRAM buffers only with incrementing addresses (that is, it cannot start at the top of a buffer, nor does it handle circular buffers automatically). The controller does not handle chaining (that is, automatically starting a new DMA transfer when one finishes).

7.8.1 CONFIGURATION

The DMA Controller is enabled via the ACTIVATE bit in DMA Main Control Register register.

Each DMA Channel must also be individually enabled via the CHANNEL_ACTIVATE bit in the DMA Channel N Activate Register to be operational.

Before starting a DMA transaction on a DMA Channel the host must assign a DMA Master to the channel via HARDWARE_FLOW_CONTROL_DEVICE. The host must not configure two different channels to the same DMA Master at the same time.

Data will be transfered between the DMA Master, starting at the programmed DEVICE_ADDRESS, and the targeted memory location, starting at the MEMORY_START_ADDRESS. The address for either the DMA Master or the targeted memory location may remain static or it may increment. To enable the DMA Master to increment its address set the INCREMENT_DEVICE_ADDRESS bit. To enable the targeted memory location to increment its addresses set the INCREMENT_MEMORY_ADDRESS. The DMA transfer will continue as long as the target memory address being accessed is less than the MEMORY_END_ADDRESS. If the DMA Controller detects that the memory location it is attempting to access on the Target is equal to the MEMORY_END_ADDRESS it will notify the DMA Master that the transaction is done. Otherwise the Data will be transferred in packets. The size of the packet is determined by the TRANSFER_SIZE.

7.8.2 OPERATION

The DMA Controller is designed to move data from one memory location to another.

7.8.2.1 Establishing a Connection

A DMA Master will initiate a DMA Transaction by requesting access to a channel. The DMA arbiter, which evaluates each channel request using a basic round robin algorithm, will grant access to the DMA master. Once granted, the channel will hold the grant until it decides to release it, by notifying the DMA Controller that it is done.

If Firmware wants to prevent any other channels from being granted while it is active it can set the LOCK_CHANNEL bit.

7.8.2.2 Initiating a Transfer

Once a connection is established the DMA Master will issue a DMA request to start a DMA transfer. If Firmware wants to have a transfer request serviced it must set the RUN bit to have its transfer requests serviced.

Firmware can initiate a transaction by setting the TRANSFER_GO bit. The DMA transfer will remain active until either the Master issues a Terminate or the DMA Controller signals that the transfer is DONE. Firmware may terminate a transaction by setting the TRANSFER_ABORT bit.

Note:	Before initiating a DMA transaction via firmware the hardware flow control must be disabled via the DISABLE_HARDWARE_FLOW_CONTROL bit.

Data may be moved from the DMA Master to the targeted Memory address or from the targeted Memory Address to the DMA Master. The direction of the transfer is determined by the TRANSFER_DIRECTION bit.

Once a transaction has been initiated firmware can use the STATUS_DONE bit to determine when the transaction is completed. This status bit is routed to the interrupt interface. In the same register there are additional status bits that indicate if the transaction completed successfully or with errors. These bits are OR'd together with the STATUS_DONE bit to generate the interrupt event. Each status be may be individually enabled/disabled from generating this event.

7.8.2.3 Reusing a DMA Channel

After a DMA Channel controller has completed, firmware **must** clear both the DMA Channel N Control Register and the DMA Channel N Interrupt Status Register. After both have been cleared to 0, the Channel Control Register can then be configured for the next transaction.

7.8.2.4 CRC Generation

A CRC generator can be attached to a DMA channel in order to generate a CRC on the data as it is transfered from the source to the destination. The CRC used is the CRC-32 algorithm used in IEEE 802.3 and many other protocols, using the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. The CRC generation takes

17.9 Description

The UART is compatible with the 16450, the 16450 ACE registers and the 16C550A. The UART performs serial-to-parallel conversions on received characters and parallel-to-serial conversions on transmit characters. Two sets of baud rates are provided. When the 1.8432 MHz source clock is selected, standard baud rates from 50 to 115.2K are available. When the source clock is 24 MHz, baud rates up to 1,500K are available. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. The UART contains a programmable baud rate generator that is capable of dividing the input clock signal by 1 to 65535. The UART is also capable of supporting the MIDI data rate. Refer to the Configuration Registers for information on disabling, powering down and changing the base address of the UART. The UART interrupt is enabled by programming OUT2 of the UART to logic “1.” Because OUT2 is logic “0,” it disables the UART's interrupt. The UART is accessible by both the Host and the EC.

17.9.1 PROGRAMMABLE BAUD RATE

The Serial Port contains a programmable Baud Rate Generator that is capable of dividing the internal clock source by any divisor from 1 to 65535. Unless an external clock source is configured, the clock source is either the 1.8432MHz clock source or the 24MHz clock source. The output frequency of the Baud Rate Generator is 16x the Baud rate. Two eight bit latches store the divisor in 16 bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16 bit Baud counter is immediately loaded. This prevents long counts on initial load. If a 0 is loaded into the BRG registers, the output divides the clock by the number 3. If a 1 is loaded, the output is the inverse of the input oscillator. If a two is loaded, the output is a divide by 2 signal with a 50% duty cycle. If a 3 or greater is loaded, the output is low for 2 bits and high for the remainder of the count.

The following tables show possible baud rates.

TABLE 17-8: UART BAUD RATES USING CLOCK SOURCE 1.8432MHz

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
50	0	2304
75	0	1536
110	0	1047
134.5	0	857
150	0	768
300	0	384
600	0	192
1200	0	96
1800	0	64
2000	0	58
2400	0	48
3600	0	32
4800	0	24
7200	0	16
9600	0	12
19200	0	6
38400	0	3
57600	0	2
115200	0	1

18.0 GPIO INTERFACE

18.1 Overview

The MEC170x GPIO interface provides general purpose input monitoring and output control, as well as managing many aspects of pin functionality; including, multi-function Pin Multiplexing Control, GPIO Direction control, Pull-up and Pull-down resistors, asynchronous wakeup and synchronous interrupt detection and Polarity control, as well as control of pin drive strength and slew rate.

Features of the GPIO interface include:

- Inputs:
  - Asynchronous rising and falling edge wakeup detection
  - Interrupt High or Low Level
- On Output:
  - Push Pull or Open Drain output
- Pull up or pull down resistor control
- Interrupt and wake capability available for all GPIOs
- Programmable pin drive strength and slew rate limiting
- Group- or individual control of GPIO data.
- Multiplexing of all multi-function pins are controlled by the GPIO interface

18.2 Power, Clocks and Reset

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

18.2.1 POWER DOMAINS

TABLE 18-1: POWER SOURCES

Name	Description
VTR	The I/O power for a subset of GPIOs is powered by this supply voltage.It may be 3.3V or 1.8V.

18.2.2 CLOCK INPUTS

TABLE 18-2: CLOCK INPUTS

Name	Description
48MHz	This clock domain is used for synchronizing GPIO inputs.

18.2.3 RESETS

TABLE 18-3: RESET SIGNALS

Name	Description
RESET_SYS	This reset is asserted when VTR is applied.
VCC_PWRGD	This signal, which if present comes directly from a pin, is asserted when the main system power rail is up.

The timer can be programmed using the Clock and Event Control register to respond to the following events using the EVENT bits and the EDGE bits: rising edge of TINx, falling edge of TINx, rising and falling edge of TINx, rising edge of overflow input, falling edge of the overflow input, and the rising and falling edges of the overflow input.

TABLE 21-6: EVENT MODE OPERATIONAL SUMMARY

Item	Description
Count Source	• External signal input to TINx pin (effective edge can be selected by software) • Timer x-1 overflow
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 21-3, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 21-3, "Timer Clock Frequencies"
Count Operation	Up/Down Counter
Reload Operation	• When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh. • When the timer overflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to 0000h.
Count Start Condition	Timer Enable is set (ENABLE = 1)
Count Stop Condition	Timer Enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	When timer overflows or underflows
TINx Pin Function	Event Generation
TOUTx Pin Function	TOUT toggles each time the timer underflows/overflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• The direction of the counter is selectable via the UPDN bit. • Reload timer on underflow/overflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Pulse Output Function The TOUTx pin changes polarity each time the timer underflows or overflows.

21.10.6.1 Event Mode Operation

The timer starts counting events when the ENABLE bit in the Timer Control Register is set and continues to count until the ENABLE bit is cleared. When the ENABLE bit is set, the timer continues counting from the current value in the timer except after a reset event. After a reset event, the timer always starts counting from the value programmed in the Reload Register if counting down or from 0000h if counting up. Figure 21-7 shows an example of timer operation in Event mode. The RLOAD bit controls the behavior of the timer when it underflows or overflows.

27.0 TACH

27.1 Introduction

This block monitors TACH output signals (or locked rotor signals) from various types of fans, and determines their speed.

27.2 References

No references have been cited for this feature.

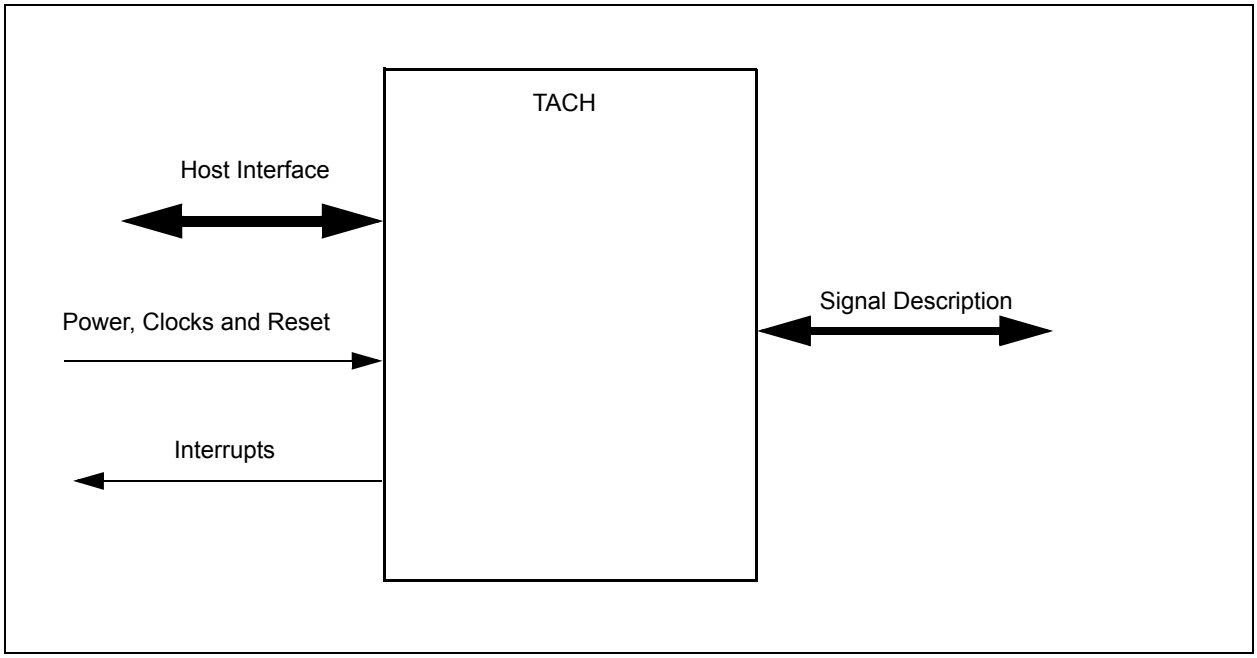
27.3 Terminology

There is no terminology defined for this section.

27.4 Interface

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

FIGURE 27-1: I/O DIAGRAM OF BLOCK



27.5 Signal Description

TABLE 27-1: SIGNAL DESCRIPTION

Name	Direction	Description
TACH INPUT	Input	Tachometer signal from TACHx Pin.

27.6 Host Interface

The registers defined for the TACH are accessible by the various hosts as indicated in Section 27.11, "EC Registers".

30.11.6 ADC CHANNEL READING REGISTERS

All 16 ADC channels return their results into a 32-bit reading register. In each case the low 10 bits of the reading register return the result of the Analog to Digital conversion and the upper 22 bits return 0. Table 30-6, "Register Summary" shows the addresses of all the reading registers.

Note:	The ADC Channel Reading Registers access require single 16, or 32 bit reads; i.e., two 8 bit reads will not provide data coherency.

37.12.4 SPI TX_DATA REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	TX_DATA A write to this register when the Tx_Data buffer is empty (TXBE in the SPI Status Register is '1') initiates a SPI transaction. The byte written to this register will be loaded into the shift register and the TXBE flag will be asserted. This indicates that the next byte can be written into the TX_DATA register. This byte will remain in the TX_-DATA register until the SPI core has finished shifting out the previous byte. Once the shift register is empty, the hardware will load the pending byte into the shift register and once again assert the TxBE bit. The TX_DATA register must not be written when the TXBE bit is zero. Writing this register may overwrite the transmit data before it is loaded into the shift register.	R/W	0h	RESET_SYS

37.12.5 SPI RX_DATA REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	R	-	-
7:0	RX_DATA This register is used to read the value returned by the external SPI device. At the end of a byte transfer the RX_DATA register contains serial input data (valid or not) from the last transaction and the RXBF bit is set to one. This status bit indicates that the RX_DATA register has been loaded with a the serial input data. The RX_DATA register should not be read before the RXBF bit is set. The RX_DATA register must be read, clearing the RXBF status bit before writing the TX_DATA register. The data in the receive shift register is only loaded into the RX_DATA register when this bit is cleared. If a data byte is pending in the receive shift register the value will be loaded immediately into the RX_DATA register and the RXBF status flag will be asserted. Software should read the RX_-DATA register twice before starting a new transaction to make sure the RX_DATA buffer and shift register are both empty.	R/W	0h	RESET_SYS

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	WRITE_PROTECT_OUT_VALUE This bit sets the value on the WRITE PROTECT SPI Output Port if it is driven. 1=WRITE PROTECT is driven to 1 0=WRITE PROTECT is driven to 0	R/W	1h	RESET

38.11.5 QMSPI STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	R	-	-
27:24	CURRENT_DESCRIPTION_BUFFER This field shows the Description Buffer currently active. This field has no meaning if Description Buffers are not enabled.	R	0h	RESET
23:17	Reserved	R	-	-
16	TRANSFER_ACTIVE 1=A transfer is currently executing 0=No transfer currently in progress	R	0h	RESET
15	RECEIVE_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to write to a full Receive Buffer) 0=No stalls occurred	R/WC	0h	RESET
14	RECEIVE_BUFFER_REQUEST This status is asserted if the Receive Buffer reaches a high water mark established by the RECEIVE_BUFFER_TRIGGER field. 1=RECEIVE_BUFFER_COUNT is greater than or equal to RECEIVE_BUFFER_TRIGGER 0=RECEIVE_BUFFER_COUNT is less than RECEIVE_BUFFER_TRIGGER	R/WC	0h	RESET
13	RECEIVE_BUFFER_EMPTY 1=The Receive Buffer is empty 0=The Receive Buffer is not empty	R	1h	RESET
12	RECEIVE_BUFFER_FULL 1=The Receive Buffer is full 0=The Receive Buffer is not full	R	0h	RESET
11	TRANSMIT_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to read from an empty Transmit Buffer) 0=No stalls occurred	R/WC	0h	RESET

46.8.4 INTERRUPT CONTROL REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	R	-	-
0	NVIC_EN This bit enables Alternate NVIC IRQ's Vectors. The Alternate NVIC Vectors provides each interrupt event with a dedicated (direct) NVIC vector. 1=Alternate NVIC vectors enabled 0=Alternate NVIC vectors disabled	R/W	1b	RESET_SYS

46.8.5 ETM TRACE ENABLE REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	R	-	-
0	TRACE_EN This bit enables the ARM TRACE debug port (ETM/ITM). The Trace Debug pins are forced to the TRACE functions. 1=ARM TRACE port enabled 0=ARM TRACE port disabled	R/W	0b	RESET_SYS

46.8.6 DEBUG ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	R	-	-
3	DEBUG_PU_EN If this bit is set to '1b' internal pull-up resistors are automatically enabled on the appropriate debugging port wires whenever the debug port is enabled (the DEBUG_EN bit in this register is '1b' and the JTAG_RST# pin is high). The setting of DEBUG_PIN_CFG determines which pins have pull-ups enabled when the debug port is enabled.	R/W	0h	RESET_SYS

51.14 8042 Emulation CPU_Reset Timing

FIGURE 51-19: KBRST TIMING

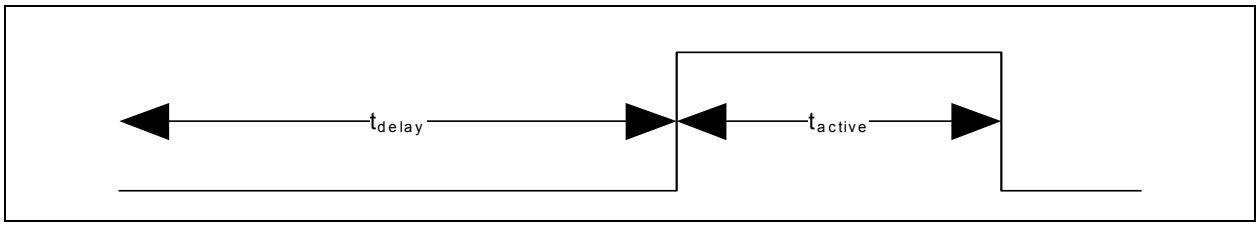


TABLE 51-16: KBRST TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{delay}	Delay prior to active pulse	14	15	15.5	µs
t_{active}	Active pulse width	6	8	8.5	µs

The KBRST pin is the CPU_RESET signal described in Section 12.11.2, "CPU_RESET Hardware Speed-Up"

TABLE 51-19: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	Period of CLK	60		302	µs
t3	Duration of CLK high (active)	30		151	
t4	Duration of CLK low (inactive)				
t5	DATA setup time to falling edge of CLK. MEC170x samples the data line on the falling CLK edge.	1			
t6	DATA hold time from falling edge of CLK. MEC170x samples the data line on the falling CLK edge.	2			
t7	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t8	Falling edge of 11th CLK to RDATA_RDY asserted.			1.6	µs
t9	Trailing edge of the EC's RD signal of the Receive Register to RDATA_RDY bit de-asserted.			500	ns
t10	Trailing edge of the EC's RD signal of the Receive Register to the CLK line released to high-Z.				
t11	PS2_CLK is "Low" and PS2_DATA is "Hi-Z" when PS2_EN is de-asserted.				
t12	RDATA_RDY asserted an interrupt is generated.				

51.18 Fan Tachometer Timing

FIGURE 51-23: FAN TACHOMETER INPUT TIMING

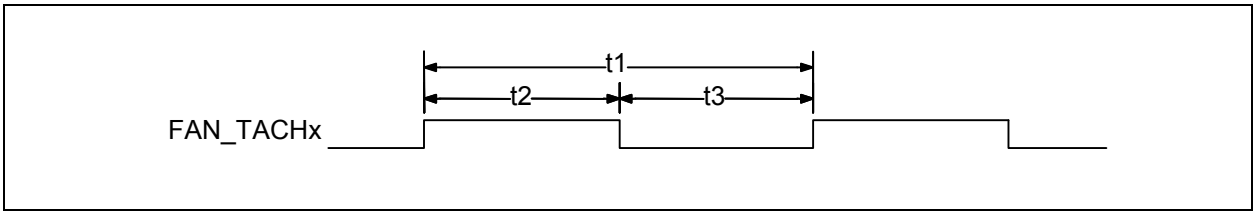


TABLE 51-21: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			µsec
t2	Pulse High Time	20			
t3	Pulse Low Time	20			
Note:	t_{TACH} is the clock used for the tachometer counter. It is 30.52 * prescaler, where the prescaler is programmed in the Fan Tachometer Timebase Prescaler register.				