

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	MIPS32® M-Class
Core Size	32-Bit Single-Core
Speed	200MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I ² C, PMP, SPI, SQI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, I ² S, POR, PWM, WDT
Number of I/O	120
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	•
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	2.1V ~ 3.6V
Data Converters	A/D 48x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-TQFP
Supplier Device Package	144-TQFP (16x16)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mz0512efk144-i-ph

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

		Pin Nu	mber				
Pin Name	64-pin QFN/ TQFP	100-pin TQFP	124-pin VTLA	144-pin TQFP/ LQFP	Pin Type	Buffer Type	Description
ERXD0	61	41	B32	81	I	ST	Ethernet Receive Data 0
ERXD1	58	42	B26	66	I	ST	Ethernet Receive Data 1
ERXD2	57	43	A31	67	I	ST	Ethernet Receive Data 2
ERXD3	56	44	A40	82	I	ST	Ethernet Receive Data 3
ERXERR	64	35	A30	65	I	ST	Ethernet Receive Error Input
ERXDV	62	12	B40	101	I	ST	Ethernet Receive Data Valid
ERXCLK	63	16	B12	27	I	ST	Ethernet Receive Clock
ETXD0	2	86	A5	7	0	—	Ethernet Transmit Data 0
ETXD1	3	85	B4	8	0	—	Ethernet Transmit Data 1
ETXD2	43	79	B17	43	0	—	Ethernet Transmit Data 2
ETXD3	46	80	A22	44	0	—	Ethernet Transmit Data 3
ETXERR	50	87	B44	114	0	—	Ethernet Transmit Error
ETXEN	1	77	A57	120	0	—	Ethernet Transmit Enable
ETXCLK	51	78	B47	121	I	ST	Ethernet Transmit Clock
ECOL	44	10	B33	83	I	ST	Ethernet Collision Detect
ECRS	45	11	A47	100	I	ST	Ethernet Carrier Sense
EMDC	30	70	B39	99	0		Ethernet Management Data Clock
EMDIO	49	71	A55	115	I/O		Ethernet Management Data
Leaend:	CMOS = CI	MOS-comp	atible input	or output		Analog =	Analog input P = Power

TABLE 1-16: ETHERNET MII I/O DESCRIPTIONS

Legend: CMOS = CMOS-compatible input or output ST = Schmitt Trigger input with CMOS levels TTL = Transistor-transistor Logic input buffer

Analog = Analog input O = Output

I = Input

PPS = Peripheral Pin Select

TABLE 1-17: ETHERNET RMII PINOUT I/O DESCRIPTIONS

		Pin Nu	mber					
Pin Name	64-pin QFN/ TQFP	100-pin TQFP	124-pin VTLA	144-pin TQFP/ LQFP	Pin Type	Buffer Type	Description	
				E	thernet I	MII Interfac	e	
ERXD0	61	41	B32	81	I	ST	Ethernet Receive Data 0	
ERXD1	58	42	B26	66	I	ST	Ethernet Receive Data 1	
ERXERR	64	35	A30	65	I	ST	Ethernet Receive Error Input	
ETXD0	2	86	A5	7	0	_	Ethernet Transmit Data 0	
ETXD1	3	85	B4	8	0	_	Ethernet Transmit Data 1	
ETXEN	1	77	A57	120	0	_	Ethernet Transmit Enable	
EMDC	30	70	B39	99	0	_	Ethernet Management Data Clock	
EMDIO	49	71	A55	115	I/O	_	Ethernet Management Data	
EREFCLK	63	16	B12	27	I	ST	Ethernet Reference Clock	
ECRSDV	62	12	B40	101	Ι	ST	Ethernet Carrier Sense Data Valid	
Legend: CMOS = CMOS-compatible input or output Analog = Analog input P = Power								

Legend: CMOS = CMOS-compatible input or output ST = Schmitt Trigger input with CMOS levels TTL = Transistor-transistor Logic input buffer Analog = Analog inputP = PowerO = OutputI = InputPPS = Peripheral Pin Select

3.1 Architecture Overview

The MIPS32 M-Class Microprocessor core in PIC32MZ EF family devices contains several logic blocks working together in parallel, providing an efficient high-performance computing engine. The following blocks are included with the core:

- Execution unit
- General Purpose Register (GPR)
- Multiply/Divide Unit (MDU)
- System control coprocessor (CP0)
- Floating Point Unit (FPU)
- Memory Management Unit (MMU)
- Instruction/Data cache controllers
- Power Management
- Instructions and data caches
- microMIPS support
- Enhanced JTAG (EJTAG) controller

3.1.1 EXECUTION UNIT

The processor core execution unit implements a load/ store architecture with single-cycle ALU operations (logical, shift, add, subtract) and an autonomous multiply/divide unit. The core contains thirty-two 32-bit General Purpose Registers (GPRs) used for integer operations and address calculation. Seven additional register file shadow sets (containing thirty-two registers) are added to minimize context switching overhead during interrupt/exception processing. The register file consists of two read ports and one write port and is fully bypassed to minimize operation latency in the pipeline.

The execution unit includes:

- 32-bit adder used for calculating the data address
- Address unit for calculating the next instruction address
- Logic for branch determination and branch target address calculation
- · Load aligner
- Trap condition comparator
- Bypass multiplexers used to avoid stalls when executing instruction streams where data producing instructions are followed closely by consumers of their results

- Leading Zero/One detect unit for implementing the CLZ and CLO instructions
- Arithmetic Logic Unit (ALU) for performing arithmetic and bitwise logical operations
- Shifter and store aligner
- DSP ALU and logic block for performing DSP instructions, such as arithmetic/shift/compare operations

3.1.2 MULTIPLY/DIVIDE UNIT (MDU)

The processor core includes a Multiply/Divide Unit (MDU) that contains a separate pipeline for multiply and divide operations, and DSP ASE multiply instructions. This pipeline operates in parallel with the Integer Unit (IU) pipeline and does not stall when the IU pipeline stalls. This allows MDU operations to be partially masked by system stalls and/or other integer unit instructions.

The high-performance MDU consists of a 32x32 booth recoded multiplier, four pairs of result/accumulation registers (HI and LO), a divide state machine, and the necessary multiplexers and control logic. The first number shown ('32' of 32x32) represents the *rs* operand. The second number ('32' of 32x32) represents the *rt* operand.

The MDU supports execution of one multiply or multiply-accumulate operation every clock cycle.

Divide operations are implemented with a simple 1-bitper-clock iterative algorithm. An early-in detection checks the sign extension of the dividend (*rs*) operand. If *rs* is 8 bits wide, 23 iterations are skipped. For a 16-bit wide *rs*, 15 iterations are skipped and for a 24-bit wide *rs*, 7 iterations are skipped. Any attempt to issue a subsequent MDU instruction while a divide is still active causes an IU pipeline stall until the divide operation has completed.

Table 3-1 lists the repeat rate (peak issue rate of cycles until the operation can be reissued) and latency (number of cycles until a result is available) for the processor core multiply and divide instructions. The approximate latency and repeat rates are listed in terms of pipeline clocks.

Opcode	Operand Size (mul rt) (div rs)	Latency	Repeat Rate
MULT/MULTU, MADD/MADDU,	16 bits	5	1
MSUB/MSUBU (HI/LO destination)	32 bits	5	1
MUL (GPR destination)	16 bits	5	1
	32 bits	5	1
DIV/DIVU	8 bits	12/14	12/14
	16 bits	20/22	20/22
	24 bits	28/30	28/30
	32 bits	36/38	36/38

TABLE 3-1:MIPS32[®] M-CLASS MICROPROCESSOR CORE HIGH-PERFORMANCE INTEGER
MULTIPLY/DIVIDE UNIT LATENCIES AND REPEAT RATES

TABLE 4-17: SYSTEM BUS TARGET 9 REGISTER MAP

ess											Bits								
Virtual Addr (BF8F_#)	Register Name	Bit Range	31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Resets
A 420		31:16	MULTI	—		—		CODE	<3:0>		_	-	—	—	—	—	—	—	0000
A420	SBIGELOGI	15:0				INI	TID<7:0>					REGIO	N<3:0>		—	С	MD<2:0>		0000
A 4 2 4		31:16		_	_	_	_	—	_				_	—	—	—	_	—	0000
A424	3B19ELOG2	15:0		_		_	_	—	—				—	—	_	_	GROU	P<1:0>	0000
A 4 2 9	SPTOFCON	31:16	_	-		_	_	-	_	ERRP			_	_	—	-	_	-	0000
A420	SBISECON	15:0	_	-		_	_	-	_				_	_	—	-	_	-	0000
A 420		31:16		_	-	—	—	—	—	-	_	_	—	—	—	_	—	—	0000
7430	OBTIECENO	15:0	—	—	_	—	—	—	—	_	_	_	—	—	—	_	—	CLEAR	0000
A/38		31:16	—	—	_	—	—	—	—	_	_	_	—	—	—	_	—	—	0000
7430	ODISECEN	15:0	—		_	—	—	_	—	—	—	_	—	—	—	_	—	CLEAR	0000
A440	SBT9REGO	31:16								BA	SE<21:6>								xxxx
71440	OBTOREGO	15:0			BA	ASE<5:0>			PRI	_	SIZE<4:0>				_	—	_	xxxx	
A450	SBT9RD0	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
//100	CETORES	15:0	_	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx
A458	SBT9WR0	31:16	_	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
//100	CETOTING	15:0		—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx
A460	SBT9REG1	31:16	31:16 BASE<21:6>								xxxx								
///00	OBTOREOT	15:0 BASE<5:0>			PRI	_			SIZE<4:0	>	-	_	—	—	xxxx				
A470	SBT9RD1	31:16	—	_	_	—	—	—	—	_	_	_	—	—	—	—	—	—	xxxx
	A470 SBT9RD1	15:0	_	_	—	—	-	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx
A478	478 SBT9W/R1 31	31:16	—	_	_	—	—	—	—	_	_	_		_	—	—	_	—	xxxx
/ 110	00100000	15:0	—	—	_	_	_	—	—	—	—	_	—	-	GROUP3	GROUP2	GROUP1	GROUP0	xxxx

PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Family

Legend: x = unknown value on Reset; - = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note: For reset values listed as 'xxxx', please refer to Table 4-6 for the actual reset values.

(1)		IRQ			Interru	า	Persistent	
Interrupt Source ^(*)	XC32 Vector Name	#	Vector #	Flag	Enable	Priority	Sub-priority	Interrupt
DMA Channel 0	_DMA0_VECTOR	134	OFF134<17:1>	IFS4<6>	IEC4<6>	IPC33<20:18>	IPC33<17:16>	No
DMA Channel 1	_DMA1_VECTOR	135	OFF135<17:1>	IFS4<7>	IEC4<7>	IPC33<28:26>	IPC33<25:24>	No
DMA Channel 2	_DMA2_VECTOR	136	OFF136<17:1>	IFS4<8>	IEC4<8>	IPC34<4:2>	IPC34<1:0>	No
DMA Channel 3	_DMA3_VECTOR	137	OFF137<17:1>	IFS4<9>	IEC4<9>	IPC34<12:10>	IPC34<9:8>	No
DMA Channel 4	_DMA4_VECTOR	138	OFF138<17:1>	IFS4<10>	IEC4<10>	IPC34<20:18>	IPC34<17:16>	No
DMA Channel 5	_DMA5_VECTOR	139	OFF139<17:1>	IFS4<11>	IEC4<11>	IPC34<28:26>	IPC34<25:24>	No
DMA Channel 6	_DMA6_VECTOR	140	OFF140<17:1>	IFS4<12>	IEC4<12>	IPC35<4:2>	IPC35<1:0>	No
DMA Channel 7	_DMA7_VECTOR	141	OFF141<17:1>	IFS4<13>	IEC4<13>	IPC35<12:10>	IPC35<9:8>	No
SPI2 Fault	_SPI2_FAULT_VECTOR	142	OFF142<17:1>	IFS4<14>	IEC4<14>	IPC35<20:18>	IPC35<17:16>	Yes
SPI2 Receive Done	_SPI2_RX_VECTOR	143	OFF143<17:1>	IFS4<15>	IEC4<15>	IPC35<28:26>	IPC35<25:24>	Yes
SPI2 Transfer Done	_SPI2_TX_VECTOR	144	OFF144<17:1>	IFS4<16>	IEC4<16>	IPC36<4:2>	IPC36<1:0>	Yes
UART2 Fault	_UART2_FAULT_VECTOR	145	OFF145<17:1>	IFS4<17>	IEC4<17>	IPC36<12:10>	IPC36<9:8>	Yes
UART2 Receive Done	_UART2_RX_VECTOR	146	OFF146<17:1>	IFS4<18>	IEC4<18>	IPC36<20:18>	IPC36<17:16>	Yes
UART2 Transfer Done	_UART2_TX_VECTOR	147	OFF147<17:1>	IFS4<19>	IEC4<19>	IPC36<28:26>	IPC36<25:24>	Yes
I2C2 Bus Collision Event ⁽²⁾	_I2C2_BUS_VECTOR	148	OFF148<17:1>	IFS4<20>	IEC4<20>	IPC37<4:2>	IPC37<1:0>	Yes
I2C2 Slave Event ⁽²⁾	_I2C2_SLAVE_VECTOR	149	OFF149<17:1>	IFS4<21>	IEC4<21>	IPC37<12:10>	IPC37<9:8>	Yes
I2C2 Master Event ⁽²⁾	_I2C2_MASTER_VECTOR	150	OFF150<17:1>	IFS4<22>	IEC4<22>	IPC37<20:18>	IPC37<17:16>	Yes
Control Area Network 1	_CAN1_VECTOR	151	OFF151<17:1>	IFS4<23>	IEC4<23>	IPC37<28:26>	IPC37<25:24>	Yes
Control Area Network 2	_CAN2_VECTOR	152	OFF152<17:1>	IFS4<24>	IEC4<24>	IPC38<4:2>	IPC38<1:0>	Yes
Ethernet Interrupt	_ETHERNET_VECTOR	153	OFF153<17:1>	IFS4<25>	IEC4<25>	IPC38<12:10>	IPC38<9:8>	Yes
SPI3 Fault	_SPI3_FAULT_VECTOR	154	OFF154<17:1>	IFS4<26>	IEC4<26>	IPC38<20:18>	IPC38<17:16>	Yes
SPI3 Receive Done	_SPI3_RX_VECTOR	155	OFF155<17:1>	IFS4<27>	IEC4<27>	IPC38<28:26>	IPC38<25:24>	Yes
SPI3 Transfer Done	_SPI3_TX_VECTOR	156	OFF156<17:1>	IFS4<28>	IEC4<28>	IPC39<4:2>	IPC39<1:0>	Yes
UART3 Fault	_UART3_FAULT_VECTOR	157	OFF157<17:1>	IFS4<29>	IEC4<29>	IPC39<12:10>	IPC39<9:8>	Yes
UART3 Receive Done	_UART3_RX_VECTOR	158	OFF158<17:1>	IFS4<30>	IEC4<30>	IPC39<20:18>	IPC39<17:16>	Yes
UART3 Transfer Done	_UART3_TX_VECTOR	159	OFF159<17:1>	IFS4<31>	IEC4<31>	IPC39<28:26>	IPC39<25:24>	Yes
I2C3 Bus Collision Event	_I2C3_BUS_VECTOR	160	OFF160<17:1>	IFS5<0>	IEC5<0>	IPC40<4:2>	IPC40<1:0>	Yes
I2C3 Slave Event	_I2C3_SLAVE_VECTOR	161	OFF161<17:1>	IFS5<1>	IEC5<1>	IPC40<12:10>	IPC40<9:8>	Yes

TABLE 7-2: INTERRUPT IRQ, VECTOR, AND BIT LOCATION (CONTINUED)

Note 1: Not all interrupt sources are available on all devices. See TABLE 1: "PIC32MZ EF Family Features" for the list of available peripherals.

2: This interrupt source is not available on 64-pin devices.

3: This interrupt source is not available on 100-pin devices.

4: This interrupt source is not available on 124-pin devices.

TABLE 7-3: INTERRUPT REGISTER MAP (CONTINUED)

ress ()		e								В	its								s									
Virtual Add (BF81 #	Register Name ⁽¹⁾	Bit Rang	31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Reset									
0550	055047	31:16	_	—	-	—	-	—	—	—	—	—	—	—	—	—	VOFF<	17:16>	0000									
05FC	OFF047	15:0		•	•		•			VOFF<15:1>	•							—	0000									
0600		31:16	_	—	_	_	—	_	_	_	—	_	—	—	—		VOFF<	17:16>	0000									
0000	011040	15:0		-		-				VOFF<15:1>	•		-	-			_	_	0000									
0604	OFF049	31:16	_	—	_	_	—	_	_	—	_	—	—		—	_	VOFF<	17:16>	0000									
0004	011040	15:0								VOFF<15:1>	•							—	0000									
0608	OFF050	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VOFF<	17:16>	0000									
	0.1.000	15:0								VOFF<15:1>	, 						1	_	0000									
0600	OFF051	31:16		—	—	—	—	_	_	—	—	—	—	—	—	—	VOFF<	17:16>	0000									
		15:0		i	-	i	i	1	1	VOFF<15:1>	, i	1	i	i	1		r	—	0000									
0610	OFF052	31:16		_	—	_	—	_	_	_	_	_	_	_	_	—	VOFF<	17:16>	0000									
		15:0								VOFF<15:1>	, 						1/055		0000									
0614	OFF053	31:16		_	_	_	—	_		-	_		_	_	_	_	VOFF<	17:16>	0000									
		15:0								VOFF<15:1>	> 						VOFF	-	0000									
0618	OFF054	31:16		_	_	_	_	_	_	VOEE -15:1>	_	_	_		—	_	VUFF<	17:16>	0000									
		31.16			_		_			VOFF<15.12							VOFE	17:16	0000									
061C	OFF055	15.0								VOFE<15:1							VOITS		0000									
		31.16	_	_	_	_	_	_	_	_	_	_	_	_	_	_	VOFF<	17:16>	0000									
0620	OFF056	15:0								VOFF<15:1>	•							_	0000									
		31:16		_	_	_	_	_	_	_	_	_	_	_	_	_	VOFF<	17:16>	0000									
0624	OFF057	15:0								VOFF<15:1>	•							_	0000									
		31:16	_	—	_		—	_	_	_	_	_	—	_	_	_	VOFF<	17:16>	0000									
0628	OFF058	15:0			•		•			VOFF<15:1>	•							_	0000									
0000	055050	31:16	_	—	_	_	_	_	_	—	_	—	_	—	_	—	VOFF<	17:16>	0000									
0620	OFF059	15:0								VOFF<15:1>	•							—	0000									
0620	OFFORD	31:16	_	—	—	_	—	_	_	_	—	_	—	—	_	-	VOFF<	17:16>	0000									
0030		15:0								VOFF<15:1>								_	0000									
0624	OFE061	31:16	_	—	-	—	-	—	-	—	—	—	—	—	—	—	VOFF<	17:16>	0000									
0034		15:0 VOFF<15:1> — 0000																										
Lege	nd: v=	inknow	n value on F	Reset: — = u	nimplemente	d read as '0	' Reset value	s are shown i	n hexadecima	1																		

DS60001320D-page 134

x = unknown value on Reset; --- = unimplemented, read as '0'. Reset values are shown in hexadecimal.

All registers in this table with the exception of the OFFx registers, have corresponding CLR, SET, and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See Section 12.3 "CLR, SET, and INV Note 1: **Registers**" for more information. This bit or register is not available on 64-pin devices.

2:

3: This bit or register is not available on devices without a CAN module.

4: This bit or register is not available on 100-pin devices.

5: Bits 31 and 30 are not available on 64-pin and 100-pin devices; bits 29 through 14 are not available on 64-pin devices.

6: Bits 31, 30, 29, and bits 5 through 0 are not available on 64-pin and 100-pin devices; bit 31 is not available on 124-pin devices; bit 22 is not available on 64-pin devices.

7: This bit or register is not available on devices without a Crypto module.

This bit or register is not available on 124-pin devices. 8:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
31.24	IFS31	IFS30	IFS29	IFS28	IFS27	IFS26	IFS25	IFS24
22:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23.10	IFS23	IFS22	IFS21	IFS20	IFS19	IFS18	IFS17	IFS16
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15.6	IFS15	IFS14	IFS13	IFS12	IFS11	IFS10	IFS9	IFS8
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7:0	IFS7	IFS6	IFS5	IFS4	IFS3	IFS2	IFS1	IFS0

REGISTER 7-5: IFSx: INTERRUPT FLAG STATUS REGISTER

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, rea	ad as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-0 IFS31-IFS0: Interrupt Flag Status bits

- 1 = Interrupt request has occurred
- 0 = No interrupt request has occurred

Note: This register represents a generic definition of the IFSx register. Refer to Table 7-2 for the exact bit definitions.

REGISTER 7-6: IECx: INTERRUPT ENABLE CONTROL REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
31.24	IEC31	IEC30	IEC29	IEC28	IEC27	IEC26	IEC25	IEC24
22.16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23.10	IEC23	IEC22	IEC21	IEC20	IEC19	IEC18	IEC17	IEC16
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15.0	IEC15	IEC14	IEC13	IEC12	IEC11	IEC10	IEC9	IEC8
7.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7:0	IEC7	IEC6	IEC5	IEC4	IEC3	IEC2	IEC1	IEC0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'				
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown			

bit 31-0 IEC31-IEC0: Interrupt Enable bits

1 = Interrupt is enabled

0 = Interrupt is disabled

Note: This register represents a generic definition of the IECx register. Refer to Table 7-2 for the exact bit definitions.

12.4.5 OUTPUT MAPPING

In contrast to inputs, the outputs of the PPS options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPnR registers (Register 12-2) are used to control output mapping. Like the [*pin name*]R registers, each register contains sets of 4 bit fields. The value of the bit field corresponds to one of the peripherals, and that peripheral's output is mapped to the pin (see Table 12-3 and Figure 12-3).

A null output is associated with the output register reset value of '0'. This is done to ensure that remappable outputs remain disconnected from all output pins by default.

FIGURE 12-3: EXAMPLE OF MULTIPLEXING OF REMAPPABLE OUTPUT FOR RPF0



12.4.6 CONTROLLING CONFIGURATION CHANGES

Because peripheral remapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. PIC32MZ EF devices include two features to prevent alterations to the peripheral map:

- Control register lock sequence
- Configuration bit select lock

12.4.6.1 Control Register Lock

Under normal operation, writes to the RPnR and [*pin name*]R registers are not allowed. Attempted writes appear to execute normally, but the contents of the registers remain unchanged. To change these registers, they must be unlocked in hardware. The register lock is controlled by the IOLOCK Configuration bit (CFGCON<13>). Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes.

To set or clear the IOLOCK bit, an unlock sequence must be executed. Refer to **Section 42. "Oscillators with Enhanced PLL"** in the *"PIC32 Family Reference Manual"* for details.

12.4.6.2 Configuration Bit Select Lock

As an additional level of safety, the device can be configured to prevent more than one write session to the RPnR and [*pin name*]R registers. The IOL1WAY Configuration bit (DEVCFG3<29>) blocks the IOLOCK bit from being cleared after it has been set once. If IOLOCK remains set, the register unlock procedure does not execute, and the PPS control registers cannot be written to. The only way to clear the bit and reenable peripheral remapping is to perform a device Reset.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31:24		—	—	—	—	—	—	—
22.16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23:16	—	—	—	—	—	—	—	—
45.0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
15:8	ON	—	SIDL	—	—	—	—	—
7.0	U-0	U-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7:0		_	OC32	OCFLT ⁽¹⁾	OCTSEL ⁽²⁾		OCM<2:0>	

REGISTER 18-1: OCxCON: OUTPUT COMPARE 'x' CONTROL REGISTER

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ead as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-16 Unimplemented: Read as '0'

- bit 15 **ON:** Output Compare Peripheral On bit
 - 1 = Output Compare peripheral is enabled
 - 0 = Output Compare peripheral is disabled
- bit 14 Unimplemented: Read as '0'
- bit 13 SIDL: Stop in Idle Mode bit
 - 1 = Discontinue operation when CPU enters Idle mode
 - 0 = Continue operation in Idle mode
- bit 12-6 Unimplemented: Read as '0'
- bit 5 **OC32:** 32-bit Compare Mode bit
 - 1 = OCxR<31:0> and/or OCxRS<31:0> are used for comparisons to the 32-bit timer source
 - 0 = OCxR<15:0> and OCxRS<15:0> are used for comparisons to the 16-bit timer source
- bit 4 OCFLT: PWM Fault Condition Status bit⁽¹⁾
 - 1 = PWM Fault condition has occurred (cleared in HW only)
 - 0 = No PWM Fault condition has occurred
- bit 3 OCTSEL: Output Compare Timer Select bit⁽²⁾
 - 1 = Timery is the clock source for this Output Compare module
 - 0 = Timerx is the clock source for this Output Compare module
- bit 2-0 OCM<2:0>: Output Compare Mode Select bits
 - 111 = PWM mode on OCx; Fault pin is enabled
 - 110 = PWM mode on OCx; Fault pin is disabled
 - 101 = Initialize OCx pin low; generate continuous output pulses on OCx pin
 - 100 = Initialize OCx pin low; generate single output pulse on OCx pin
 - 011 = Compare event toggles OCx pin
 - 010 = Initialize OCx pin high; compare event forces OCx pin low
 - 001 = Initialize OCx pin low; compare event forces OCx pin high
 - 000 = Output compare peripheral is disabled but continues to draw current
- **Note 1:** This bit is only used when OCM<2:0> = '111'. It is read as '0' in all other modes.
 - **2:** Refer to Table 18-1 for Timerx and Timery selections.

REGISTER 19-1: SPIxCON: SPI CONTROL REGISTER (CONTINUED)

- bit 5 MSTEN: Master Mode Enable bit
 - 1 = Master mode
 - 0 = Slave mode
- bit 4 **DISSDI:** Disable SDI bit⁽⁴⁾
 - 1 = SDI pin is not used by the SPI module (pin is controlled by PORT function)
 - 0 = SDI pin is controlled by the SPI module
- bit 3-2 STXISEL<1:0>: SPI Transmit Buffer Empty Interrupt Mode bits
 - 11 = Interrupt is generated when the buffer is not full (has one or more empty elements)
 - 10 = Interrupt is generated when the buffer is empty by one-half or more
 - 01 = Interrupt is generated when the buffer is completely empty
 - 00 = Interrupt is generated when the last transfer is shifted out of SPISR and transmit operations are complete
- bit 1-0 SRXISEL<1:0>: SPI Receive Buffer Full Interrupt Mode bits
 - 11 = Interrupt is generated when the buffer is full
 - 10 = Interrupt is generated when the buffer is full by one-half or more
 - 01 = Interrupt is generated when the buffer is not empty
 - 00 = Interrupt is generated when the last word in the receive buffer is read (i.e., buffer is empty)
- **Note 1:** This bit can only be written when the ON bit = 0. Refer to **Section 37.0 "Electrical Characteristics"** for maximum clock frequency requirements.
 - 2: This bit is not used in the Framed SPI mode. The user should program this bit to '0' for the Framed SPI mode (FRMEN = 1).
 - **3:** When AUDEN = 1, the SPI/I²S module functions as if the CKP bit is equal to '1', regardless of the actual value of the CKP bit.
 - 4: This bit present for legacy compatibility and is superseded by PPS functionality on these devices (see Section 12.4 "Peripheral Pin Select (PPS)" for more information).

TABLE 21-1: I2C1 THROUGH I2C5 REGISTER MAP (CONTINUED)

sse										В	ts								
Virtual Addr (BF82_#)	Register Name ⁽¹⁾	Bit Range	31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Resets
0420	IDCOMER	31:16	_		_	_			_			_	—	—		_			0000
0430	IZCONOR	15:0	—	_	_	_							Address Ma	ask Registe	r				0000
0440	12C3BRG	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
0.10	.2005.10	15:0							Bau	d Rate Ger	erator Reg	ister							0000
0450	I2C3TRN	31:16	_	—		—	_	_	_	_	—	—	—	-	-	—	—	—	0000
-		15:0	—	_		_	—	_	_	_		1		Iransmit	Register				0000
0460	I2C3RCV	31:16	_	_		_	_		_	_	_		_	- Dessive	—	_	_	_	0000
		15:0		_		_	_		_			DOIE		Receive	Register	CRODE			0000
0600	I2C4CON	15.0					STRICT			SMENI					SDARI PCEN	DEN		SEN	1000
		31.16									- GOLIN		ACKD1	ACKEN					0000
0610	I2C4STAT	15:0	ACKSTAT	TRSTAT	ACKTIM	_	_	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	Р	S	R/W	RBF	TBF	0000
		31:16	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	0000
0620	I2C4ADD	15:0		_	_	_	_	_					Address	Register					0000
0620		31:16	_	_	_	_	_	-	_	_	_	_	—	_	_	—	_	—	0000
0630	12041VISK	15:0	_	_	—	_							Address Ma	ask Registe	r				0000
0640		31:16	—	_	_	—	-		_	-		-	_	_		_	-	_	0000
0040	1204010	15:0			-				Bau	d Rate Ger	erator Reg	ister						-	0000
0650	I2C4TRN	31:16	_	_		_	_	_	_		_	—			_	—	_	—	0000
		15:0	—	—	_	—	—	—	—	—		-		Transmit	Register				0000
0660	I2C4RCV	31:16	_	_		_	_	_		_	_	—	—	_		_	—	—	0000
		15:0	_	_		_	_	_	_					Receive	Register				0000
0800	I2C5CON	31:16	-	_	-	-	-	—	-	-	-	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000
		15:0	ON		SIDL	SCLREL	STRICT	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
0810	I2C5STAT	15.0		TPSTAT				- BCI	- CCSTAT			-	— D/A		-		DRE	TRE	0000
		31.16						BOL		ADD10		12001							0000
0820	I2C5ADD	15:0		_	_	_	_	_					Address	Register					0000
		31:16	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	0000
0830	I2C5MSK	15:0	_		_	_	_	_					Address Ma	ask Registe	r				0000
0040	1005000	31:16	_	_	_	_	_	_	_	—	_	_	—	_	_	—	—	—	0000
0840	12C5BRG	15:0							Bau	d Rate Ger	erator Reg	ister							0000
0850		31:16			—	_	—		—	_	_		_	_	_		—		0000
0000	12001 KIN	15:0	_	_	—	—	_		_	_				Transmit	Register				0000
0860	I2C5RCV	31:16	_	_		—	_	—	-	_	—	—	—	—	—	—	—	—	0000
0000	00.00	15:0	_	—	—	—	—	—	—	—				Receive	Register				0000

PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Family

Legend: x = unknown value on Reset; - = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: All registers in this table except I2CxRCV have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See Section 12.3 "CLR, SET, and INV Registers" for more information.

2: This register is not available on 64-pin devices.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31:24	—	—	—	—	—	-	_	—
00.40	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23.10		—	—		—			—
45.0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
15:8	ON	—	SIDL	IREN	RTSMD	_	UEN<	1:0> ⁽¹⁾
7.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7:0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL	<1:0>	STSEL

REGISTER 22-1: UXMODE: UARTX MODE REGISTER

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ead as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-16 Unimplemented: Read as '0'

- bit 15 ON: UARTx Enable bit
 - 1 = UARTx is enabled. UARTx pins are controlled by UARTx as defined by UEN<1:0> and UTXEN control bits
 - UARTx is disabled. All UARTx pins are controlled by corresponding bits in the PORTx, TRISx and LATx registers; UARTx power consumption is minimal
- bit 14 Unimplemented: Read as '0'
- bit 13 **SIDL:** Stop in Idle Mode bit
 - 1 = Discontinue operation when device enters Idle mode
 - 0 = Continue operation in Idle mode
- bit 12 IREN: IrDA Encoder and Decoder Enable bit
 - 1 = IrDA is enabled
 - 0 = IrDA is disabled
- bit 11 RTSMD: Mode Selection for UxRTS Pin bit
 - $1 = \overline{\text{UxRTS}}$ pin is in Simplex mode
 - $0 = \overline{\text{UxRTS}}$ pin is in Flow Control mode
- bit 10 Unimplemented: Read as '0'

bit 9-8 UEN<1:0>: UARTx Enable bits⁽¹⁾

- 11 = UxTX, UxRX and UxBCLK pins are enabled and used; UxCTS pin is controlled by corresponding bits in the PORTx register
- 10 = UxTX, UxRX, $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins are enabled and used
- 01 = UxTX, UxRX and UxRTS pins are enabled and used; UxCTS pin is controlled by corresponding bits in the PORTx register
- 00 = UxTX and UxRX pins are enabled and used; UxCTS and UxRTS/UxBCLK pins are controlled by corresponding bits in the PORTx register
- bit 7 WAKE: Enable Wake-up on Start bit Detect During Sleep Mode bit
 - 1 = Wake-up is enabled
 - 0 = Wake-up is disabled
- bit 6 LPBACK: UARTx Loopback Mode Select bit
 - 1 = Loopback mode is enabled
 - 0 = Loopback mode is disabled
- Note 1: These bits are present for legacy compatibility, and are superseded by PPS functionality on these devices. For additional information, see Section 12.4 "Peripheral Pin Select (PPS)".

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31:24	—	—	—	—	—	—	—	—
00.40	R/W-0, HC	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
23:16	RDSTART	—	—	—	—	—	DUALBUF	—
45.0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15:8	ON	—	SIDL	ADRML	ADRMUX<1:0>		PTWREN	PTRDEN
7.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
7:0	CSF<1:0> ⁽¹⁾		ALP ⁽¹⁾	CS2P ⁽¹⁾	CS1P ⁽¹⁾	_	WRSP	RDSP

REGISTER 23-1: PMCON: PARALLEL PORT CONTROL REGISTER

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ad as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-24 Unimplemented: Read as '0'

bit 23 RDSTART: Start Read on PMP Bus bit This bit is cleared by hardware at the end of the read cycle. 1 = Start a read cycle on the PMP bus 0 = No effect
bit 22-18 Unimplemented: Read as '0'
bit 17 DUALBUF: Dual Read/Write Buffers enable bit

This bit is valid in Master mode only.

1 = PMP uses separate registers for reads and writes (PMRADDR, PMDATAIN, PMWADDR, PMDATAOUT)

0 = PMP uses legacy registers (PMADDR, PMDATA)

- bit 16 Unimplemented: Read as '0'
- bit 15 **ON:** Parallel Master Port Enable bit

1 = PMP is enabled

- 0 = PMP is disabled, no off-chip access performed
- bit 14 Unimplemented: Read as '0'
- bit 13 **SIDL:** Stop in Idle Mode bit
 - 1 = Discontinue module operation when device enters Idle mode
 - 0 = Continue module operation in Idle mode

bit 12-11 ADRMUX<1:0>: Address/Data Multiplexing Selection bits

- 11 = Lower 8 bits of address are multiplexed on PMD<15:0> pins; upper 8 bits are not used
- 10 = All 16 bits of address are multiplexed on PMD<15:0> pins
- 01 = Lower 8 bits of address are multiplexed on PMD<7:0> pins, upper bits are on PMA<15:8>
- 00 = Address and data appear on separate pins
- bit 10 PMPTTL: PMP Module TTL Input Buffer Select bit
 - 1 = PMP module uses TTL input buffers
 - 0 = PMP module uses Schmitt Trigger input buffer
- bit 9 **PTWREN:** Write Enable Strobe Port Enable bit
 - 1 = PMWR/PMENB port is enabled
 - 0 = PMWR/PMENB port is disabled
- bit 8 **PTRDEN:** Read/Write Strobe Port Enable bit
 - 1 = PMRD/PMWR port is enabled
 - 0 = PMRD/PMWR port is disabled

Note 1: These bits have no effect when their corresponding pins are used as address lines.

REGISTER 28-6: ADCIMCON2: ADC INPUT MODE CONTROL REGISTER 2 (CONTINUED)

DIFF26: AN26 Mode bit ⁽¹⁾
1 = AN26 is using Differential mode
0 = AN26 is using Single-ended mode
SIGN26: AN26 Signed Data Mode bit ⁽¹⁾
1 = AN26 is using Signed Data mode
0 = AN26 is using Unsigned Data mode
DIFF25: AN25 Mode bit ⁽¹⁾
1 = AN25 is using Differential mode
0 = AN25 is using Single-ended mode
SIGN25: AN25 Signed Data Mode bit ⁽¹⁾
1 = AN25 is using Signed Data mode
0 = AN25 is using Unsigned Data mode
DIFF24: AN24 Mode bit ⁽¹⁾
1 = AN24 is using Differential mode
0 = AN24 is using Single-ended mode
SIGN24: AN24 Signed Data Mode bit ⁽¹⁾
1 = AN24 is using Signed Data mode
0 = AN24 is using Unsigned Data mode
DIFF23: AN23 Mode bit ⁽¹⁾
1 = AN23 is using Differential mode
0 = AN23 is using Single-ended mode
SIGN23: AN23 Signed Data Mode bit ⁽¹⁾
1 = AN23 is using Signed Data mode
0 = AN23 is using Unsigned Data mode
DIFF22: AN22 Mode bit ⁽¹⁾
1 = AN22 is using Differential mode
0 = AN22 is using Single-ended mode
SIGN22: AN22 Signed Data Mode bit ⁽¹⁾
1 = AN22 is using Signed Data mode
0 = AN22 is using Unsigned Data mode
DIFF21: AN21 Mode bit ⁽¹⁾
1 = AN21 is using Differential mode
0 = AN21 is using Single-ended mode
SIGN21: AN21 Signed Data Mode bit ⁽¹⁾
1 = AN21 is using Signed Data mode
0 = AN21 is using Unsigned Data mode
DIFF20: AN20 Mode bit ⁽¹⁾
1 = AN20 is using Differential mode
0 = AN20 is using Single-ended mode
SIGN20: AN20 Signed Data Mode bit ⁽¹⁾
1 = AN20 is using Signed Data mode
0 = AN20 is using Unsigned Data mode
DIFF19: AN19 Mode bit ⁽¹⁾
1 = AN19 is using Differential mode
0 = AN19 is using Single-ended mode

Note 1: This bit is not available on 64-pin devices.

REGISTER 28-15: ADCCMPx: ADC DIGITAL COMPARATOR 'x' LIMIT VALUE REGISTER ('x' = 1 THROUGH 6)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0		
21.24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
31.24				DCMPHI<	15:8> ^(1,2,3)					
22:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
23.10	DCMPHI<7:0> ^(1,2,3)									
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
10.0	DCMPLO<15:8> ^(1,2,3)									
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
		DCMPLO<7:0> ^(1,2,3)								

Logond

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, r	read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

DCMPHI<15:0>: Digital Comparator 'x' High Limit Value bits^(1,2,3) bit 31-16 These bits store the high limit value, which is used by digital comparator for comparisons with ADC converted data.

- DCMPLO<15:0>: Digital Comparator 'x' Low Limit Value bits^(1,2,3) bit 15-0 These bits store the low limit value, which is used by digital comparator for comparisons with ADC converted data.
- Changing theses bits while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable Note 1: behavior.
 - 2: The format of the limit values should match the format of the ADC converted value in terms of sign and fractional settings.
 - 3: For Digital Comparator 0 used in CVD mode, the DCMPHI<15:0> and DCMPLO<15:0> bits must always be specified in signed format, as the CVD output data is differential and is always signed.

Table 30-1, Table 30-2, Table 30-3 and Table 30-4 show four interfaces and the associated pins that can be used with the Ethernet Controller.

TABLE 30-1: MII MODE DEFAULT INTERFACE SIGNALS (FMIIEN = 1, FETHIO = 1)

Pin Name	Description
EMDC	Management Clock
EMDIO	Management I/O
ETXCLK	Transmit Clock
ETXEN	Transmit Enable
ETXD0	Transmit Data
ETXD1	Transmit Data
ETXD2	Transmit Data
ETXD3	Transmit Data
ETXERR	Transmit Error
ERXCLK	Receive Clock
ERXDV	Receive Data Valid
ERXD0	Receive Data
ERXD1	Receive Data
ERXD2	Receive Data
ERXD3	Receive Data
ERXERR	Receive Error
ECRS	Carrier Sense
ECOL	Collision Indication

TABLE 30-2:RMII MODE DEFAULT
INTERFACE SIGNALS
(FMIIEN = 0, FETHIO = 1)

Pin Name	Description
EMDC	Management Clock
EMDIO	Management I/O
ETXEN	Transmit Enable
ETXD0	Transmit Data
ETXD1	Transmit Data
EREFCLK	Reference Clock
ECRSDV	Carrier Sense – Receive Data Valid
ERXD0	Receive Data
ERXD1	Receive Data
ERXERR	Receive Error

Note: Ethernet controller pins that are not used by selected interface can be used by other peripherals.

TABLE 30-3: MII MODE ALTERNATE INTERFACE SIGNALS (FMIIEN = 1, FETHIO = 0)

Pin Name	Description
AEMDC	Management Clock
AEMDIO	Management I/O
AETXCLK	Transmit Clock
AETXEN	Transmit Enable
AETXD0	Transmit Data
AETXD1	Transmit Data
AETXD2	Transmit Data
AETXD3	Transmit Data
AETXERR	Transmit Error
AERXCLK	Receive Clock
AERXDV	Receive Data Valid
AERXD0	Receive Data
AERXD1	Receive Data
AERXD2	Receive Data
AERXD3	Receive Data
AERXERR	Receive Error
AECRS	Carrier Sense
AECOL	Collision Indication
Nata. The	

Note: The MII mode Alternate Interface is not available on 64-pin devices.

TABLE 30-4:RMII MODE ALTERNATE
INTERFACE SIGNALS
(FMIIEN = 0, FETHIO = 0)

Pin Name	Description
AEMDC	Management Clock
AEMDIO	Management I/O
AETXEN	Transmit Enable
AETXD0	Transmit Data
AETXD1	Transmit Data
AEREFCLK	Reference Clock
AECRSDV	Carrier Sense – Receive Data Valid
AERXD0	Receive Data
AERXD1	Receive Data
AERXERR	Receive Error

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
21.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31.24	—	—	—		—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—		—	—	—	—
15.0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
15.8	—	TXBUSEIE ⁽¹⁾	RXBUSEIE ⁽²⁾	_	—	—	EWMARKIE ⁽²⁾	FWMARKIE ⁽²⁾
7:0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
7.0	RXDONEIE ⁽²⁾	PKTPENDIE ⁽²⁾	RXACTIE ⁽²⁾		TXDONEIE ⁽¹⁾	TXABORTIE ⁽¹⁾	RXBUFNAIE ⁽²⁾	RXOVFLWIE ⁽²⁾

REGISTER 30-13: ETHIEN: ETHERNET CONTROLLER INTERRUPT ENABLE REGISTER

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ad as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-15 Unimplemented: Read as '0'

- bit 14 **TXBUSEIE:** Transmit BVCI Bus Error Interrupt Enable bit⁽¹⁾
 - 1 = Enable TXBUS Error Interrupt
 - 0 = Disable TXBUS Error Interrupt
- bit 13 **RXBUSEIE:** Receive BVCI Bus Error Interrupt Enable bit⁽²⁾
 - 1 = Enable RXBUS Error Interrupt
 - 0 = Disable RXBUS Error Interrupt
- bit 12-10 Unimplemented: Read as '0'

bit 9	EWMARKIE: Empty Watermark Interrupt Enable bit ⁽²⁾ 1 = Enable EWMARK Interrupt 0 = Disable EWMARK Interrupt
bit 8	FWMARKIE: Full Watermark Interrupt Enable bit ⁽²⁾
	1 = Enable FWMARK Interrupt
	0 = Disable FWMARK Interrupt
bit 7	RXDONEIE: Receiver Done Interrupt Enable bit ⁽²⁾
	1 = Enable RXDONE Interrupt
	0 = Disable RXDONE Interrupt
bit 6	PKTPENDIE: Packet Pending Interrupt Enable bit ⁽²⁾
	1 = Enable PKTPEND Interrupt
	0 = Disable PKTPEND Interrupt
bit 5	RXACTIE: RX Activity Interrupt Enable bit
	1 = Enable RXACT Interrupt
	0 = Disable RXAC1 Interrupt
bit 4	Unimplemented: Read as '0'
bit 3	TXDONEIE: Transmitter Done Interrupt Enable bit ⁽¹⁾
	1 = Enable TXDONE Interrupt
	0 = Disable TXDONE Interrupt
bit 2	TXABORTIE: Transmitter Abort Interrupt Enable bit ⁽¹⁾
	1 = Enable TXABORT Interrupt
	0 = Disable TXABORT Interrupt
bit 1	RXBUFNAIE: Receive Buffer Not Available Interrupt Enable bit ⁽²⁾
	1 = Enable RXBUFNA Interrupt
	0 = Disable RXBUFNA Interrupt
bit 0	RXOVFLWIE: Receive FIFO Overflow Interrupt Enable bit ⁽²⁾
	1 = Enable RXOVELW Interrupt

- 0 = Disable RXOVFLW Interrupt
- **Note 1:** This bit is only used for TX operations.
 - **2:** This bit is only used for RX operations.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31.24	—	—	—	-	—	—	—	—
22:46	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23:16 BUFCNT<7:0> ⁽¹⁾				0> ⁽¹⁾				
15.0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
10.0	—	—	—	_	—	—	_	—
7:0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
	ETHBUSY ⁽⁵⁾	TXBUSY ^(2,6)	RXBUSY ^(3,6)	—	—	—	—	—

REGISTER 30-15: ETHSTAT: ETHERNET CONTROLLER STATUS REGISTER

W = Writable bit

'1' = Bit is set

Legend:

R = Readable bit

-n = Value at POR

bit 31-24 Unimplemented: Read as '0'

bit 23-16 **BUFCNT<7:0>:** Packet Buffer Count bits⁽¹⁾

Number of packet buffers received in memory. Once a packet has been successfully received, this register is incremented by hardware based on the number of descriptors used by the packet. Software decrements the counter (by writing to the BUFCDEC bit (ETHCON1<0>) for each descriptor used) after a packet has been read out of the buffer. The register does not roll over (0xFF to 0x00) when hardware tries to increment the register and the register is already at 0xFF. Conversely, the register does not roll under (0x00 to 0xFF) when software tries to decrement the register and the register is already at 0xO00. When software attempts to decrement the same time that the hardware attempts to increment the counter, the counter value will remain unchanged.

U = Unimplemented bit, read as '0'

x = Bit is unknown

'0' = Bit is cleared

When this register value reaches 0xFF, the RX logic will halt (only if automatic Flow Control is enabled) awaiting software to write the BUFCDEC bit in order to decrement the register below 0xFF.

If automatic Flow Control is disabled, the RXDMA will continue processing and the BUFCNT will saturate at a value of 0xFF.

When this register is non-zero, the PKTPEND status bit will be set and an interrupt may be generated, depending on the value of the ETHIEN bit <PKTPENDIE> register.

When the ETHRXST register is written, the BUFCNT counter is automatically cleared to 0x00.

- **Note:** BUFCNT will not be cleared when ON is set to '0'. This enables software to continue to utilize and decrement this count.
- bit 15-8 **Unimplemented:** Read as '0'
- bit 7 ETHBUSY: Ethernet Module busy bit⁽⁵⁾
 - 1 = Ethernet logic has been turned on (ON (ETHCON1<15>) = 1) or is completing a transaction 0 = Ethernet logic is idle

This bit indicates that the module has been turned on or is completing a transaction after being turned off.

- Note 1: This bit is only used for RX operations.
 - 2: This bit is only affected by TX operations.
 - 3: This bit is only affected by RX operations.
 - 4: This bit is affected by TX and RX operations.
 - 5: This bit will be set when the ON bit (ETHCON1<15>) = 1.
 - 6: This bit will be *cleared* when the ON bit (ETHCON1<15>) = 0.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
21.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31.24	—	—	—	—	—	—	—	—
22:46	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23.10	—	—	—	—	—	—	—	—
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
10.0	MWTD<15:8>							
7.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7.0				MWTD<7	:0>			

REGISTER 30-34: EMAC1MWTD: ETHERNET CONTROLLER MAC MII MANAGEMENT WRITE DATA REGISTER

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, r	ead as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-16 Unimplemented: Read as '0'

bit 15-0 MWTD<15:0>: MII Management Write Data bits

When written, a MII Management write cycle is performed using the 16-bit data and the preconfigured PHY and Register addresses from the EMAC1MADR register.

Note: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.

REGISTER 30-35: EMAC1MRDD: ETHERNET CONTROLLER MAC MII MANAGEMENT READ DATA REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
21.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31.24	—	—	—	—	—	—	_	—
22:46	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23:16	—	—	—	—	—	—	_	—
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15.0	MRDD<15:8>							
7.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7.0				MRDD	<7:0>			

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, r	ead as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-16 Unimplemented: Read as '0'

bit 15-0 MRDD<15:0>: MII Management Read Data bits

Following a MII Management Read Cycle, the 16-bit data can be read from this location.

Note: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.

36.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

36.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

36.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a highspeed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

36.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a fullspeed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming[™] (ICSP[™]).

36.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

A.3 CPU

The CPU in the PIC32MZ EF family of devices has been changed to the MIPS32 M-Class MPU architecture. This CPU includes DSP ASE, internal data and instruction L1 caches, and a TLB-based MMU.

TABLE A-4: CPU DIFFERENCES

Table A-4 summarizes some of the key differences (indicated by **Bold** type) in the internal CPU registers.

PIC32MX5XX/6XX/7XX Feature	PIC32MZ EF Feature
L1 Data and Instruction Cad	he and Prefetch Wait States
On PIC32MX devices, the cache was included in the prefetch module outside the CPU.	On PIC32MZ EF devices, the CPU has a separate L1 instruction and data cache in the core. The PREFEN<1:0> bits still enable the prefetch module; however, the K0<2:0> bits in the CP0 regis- ters controls the internal L1 cache for the designated regions.
 PREFEN<1:0> (CHECON<5:4>) 11 = Enable predictive prefetch for both cacheable and non-cacheable regions 10 = Enable predictive prefetch for non-cacheable regions only 01 = Enable predictive prefetch for cacheable regions only 00 = Disable predictive prefetch 	PREFEN<1:0> (PRECON<5:4>) 11 = Enable predictive prefetch for any address 10 = Enable predictive prefetch for CPU instructions and CPU data 01 = Enable predictive prefetch for CPU instructions only 00 = Disable predictive prefetch
DCSZ<1:0> (CHECON<9:8>) Changing these bits causes all lines to be reinitialized to the "invalid" state. 11 = Enable data caching with a size of 4 lines 10 = Enable data caching with a size of 2 lines 01 = Enable data caching with a size of 1 line 00 = Disable data caching	K0<2:0> (CP0 Reg 16, Select 0) 011 = Cacheable, non-coherent, write-back, write allocate 010 = Uncached 001 = Cacheable, non-coherent, write-through, write allocate 000 = Cacheable, non-coherent, write-through, no write allocate
CHECOH (CHECON<16>) 1 = Invalidate all data and instruction lines 0 = Invalidate all data and instruction lines that are not locked	
	The Program Flash Memory read wait state frequency points have changed in PIC32MZ EF devices. The register for accessing the PFMWS field has changed from CHECON to PRECON.
PFMWS<2:0> (CHECON<2:0>) 111 = Seven Wait states 110 = Six Wait states 101 = Five Wait states 100 = Four Wait states 011 = Three Wait states 010 = Two Wait states (61-80 MHz) 001 = One Wait state (31-60 MHz) 000 = Zero Wait state (0-30 MHz)	PFMWS<2:0> (PRECON<2:0>) 111 = Seven Wait states • • 100 = Four Wait states (200-252 MHz) 011 = Reserved 010 = Two Wait states (133-200 MHz) 001 = One Wait state (66-133 MHz) 000 = Zero Wait states (0-66 MHz)
	Note: Wait states listed are for ECC enabled.
Core Instruct On PIC32MX devices, the CPU can execute MIPS16e instructions and uses a 16-bit instruction set, which reduces memory size.	On PIC32MZ EF devices, the CPU can operate a mode called microMIPS. microMIPS mode is an enhanced MIPS32® instruction set that uses both 16-bit and 32-bit opcodes. This mode of operation reduces memory size with minimum performance impact.
	The BOOTISA (DEVCFG0<6>) Configuration bit controls the MIPS32 and microMIPS modes for boot and exception code. 1 = Boot code and Exception code is MIPS32 [®] (ISAONEXC bit is set to '0' and the ISA<1:0> bits are set to '10' in the CP0 Config3 register) 0 = Boot code and Exception code is microMIPS™ (ISAONEXC bit is set to '1' and the ISA<1:0> bits are set to '11' in the CP0 Config3 register)