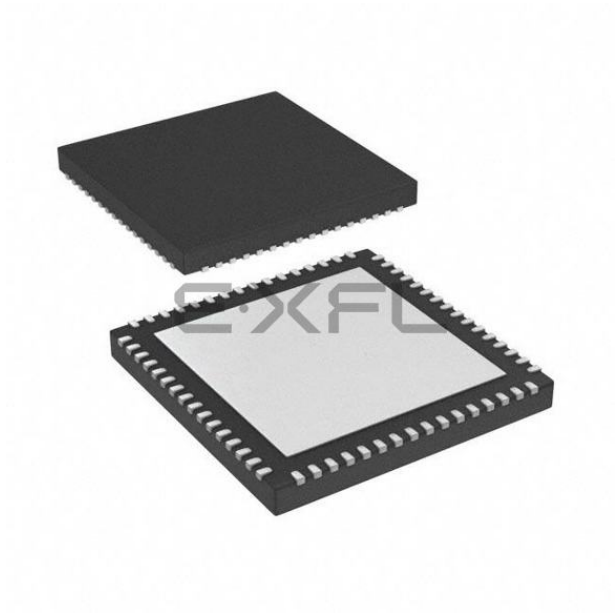


Welcome to E-XFL.COM

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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	MIPS32® M-Class
Core Size	32-Bit Single-Core
Speed	180MHz
Connectivity	Ethernet, I²C, PMP, SPI, SQI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	46
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	2.1V ~ 3.6V
Data Converters	A/D 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mz1024efg064-e-mr

1.0 DEVICE OVERVIEW

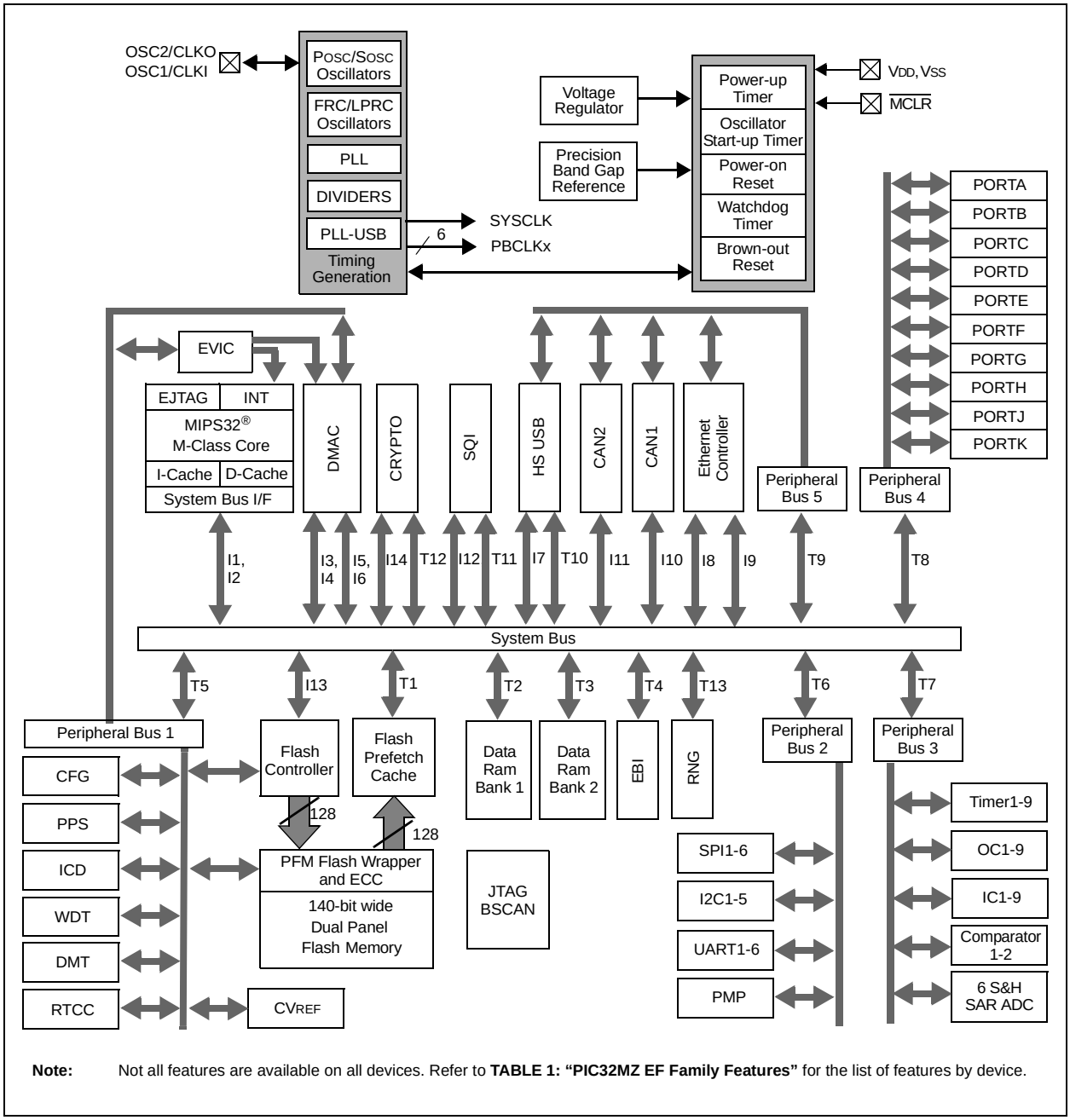
Note: This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32).

This data sheet contains device-specific information for PIC32MZ EF devices.

Figure 1-1 illustrates a general block diagram of the core and peripheral modules in the PIC32MZ EF family of devices.

Table 1-21 through Table 1-22 list the pinout I/O descriptions for the pins shown in the device pin tables (see Table 2 through Table 5).

FIGURE 1-1: PIC32MZ EF FAMILY BLOCK DIAGRAM



Note: Not all features are available on all devices. Refer to **TABLE 1: "PIC32MZ EF Family Features"** for the list of features by device.

3.0 CPU

Note 1: This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 50. "CPU for Devices with MIPS32® microAptiv™ and M-Class Cores"** (DS60001192) of the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32).

2: The Series 5 Warrior M-class CPU core resources are available at: www.imgtec.com.

The MIPS32® M-Class Core is the heart of the PIC32MZ EF family device processor. The CPU fetches instructions, decodes each instruction, fetches source operands, executes each instruction and writes the results of instruction execution to the proper destinations.

Key features include:

- 5-stage pipeline
- 32-bit address and data paths
- MIPS32 Enhanced Architecture (Release 5):
  - Multiply-accumulate and multiply-subtract instructions
  - Targeted multiply instruction
  - Zero/One detect instructions
  - `WAIT` instruction
  - Conditional move instructions (`MOVN`, `MOVZ`)
  - Vectored interrupts
  - Programmable exception vector base
  - Atomic interrupt enable/disable
  - GPR shadow registers to minimize latency for interrupt handlers
  - Bit field manipulation instructions
  - Virtual memory support
- microMIPS™ compatible instruction set:
  - Improves code size density over MIPS32, while maintaining MIPS32 performance.
  - Supports all MIPS32 instructions (except branch-likely instructions)
  - Fifteen additional 32-bit instructions and 39 16-bit instructions corresponding to commonly-used MIPS32 instructions
  - Stack pointer implicit in instruction
  - MIPS32 assembly and ABI compatible
- MMU with Translation Lookaside Buffer (TLB) mechanism:
  - 16 dual-entry fully associative Joint TLB
  - 4-entry fully associative Instruction and Data TLB
  - 4 KB pages
- Separate L1 data and instruction caches:
  - 16 KB 4-way Instruction Cache (I-Cache)
  - 4 KB 4-way Data Cache (D-Cache)
- Autonomous Multiply/Divide Unit (MDU):
  - Maximum issue rate of one 32x32 multiply per clock
  - Early-in iterative divide. Minimum 12 and maximum 38 clock latency (dividend (*rs*) sign extension-dependent)
- Power Control:
  - Minimum frequency: 0 MHz
  - Low-Power mode (triggered by `WAIT` instruction)
  - Extensive use of local gated clocks
- EJTAG Debug and Instruction Trace:
  - Support for single stepping
  - Virtual instruction and data address/value breakpoints
  - Hardware breakpoint supports both address match and address range triggering.
  - Eight instruction and four data complex breakpoints
- iFlowtrace® version 2.0 support:
  - Real-time instruction program counter
  - Special events trace capability
  - Two performance counters with 34 user-selectable countable events
  - Disabled if the processor enters Debug mode
  - Program Counter sampling
- Four Watch registers:
  - Instruction, Data Read, Data Write options
  - Address match masking options
- DSP ASE Extension:
  - Native fractional format data type operations
  - Register Single Instruction Multiple Data (SIMD) operations (add, subtract, multiply, shift)
  - GPR-based shift
  - Bit manipulation
  - Compare-Pick
  - DSP Control Access
  - Indexed-Load
  - Branch
  - Multiplication of complex operands
  - Variable bit insertion and extraction
  - Virtual circular buffers
  - Arithmetic saturation and overflow handling
  - Zero-cycle overhead saturation and rounding operations
- Floating Point Unit (FPU):
  - 1985 IEEE-754 compliant Floating Point Unit
  - Supports single and double precision datatypes
  - 2008 IEEE-754 compatibility control of NaN handling and Abs/Neg instructions
  - Runs at 1:1 core/FPU clock ratio

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-bit-per-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.

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

Opcode	Operand Size (mul *rt*) (div *rs*)	Latency	Repeat Rate
`MULT/MULTU, MADD/MADDU, MSUB/MSUBU` (HI/LO destination)	16 bits	5	1
	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

3.1.4 FLOATING POINT UNIT (FPU)

The Floating Point Unit (FPU), Coprocessor (CP1), implements the MIPS Instruction Set Architecture for floating point computation. The implementation supports the ANSI/IEEE Standard 754 (IEEE for Binary Floating Point Arithmetic) for 32-bit and 64-bit floating point data formats. The FPU can be programmed to have thirty-two 32-bit or 64-bit floating point registers used for floating point operations.

The performance is optimized for 32-bit formats. Most instructions have one FPU cycle throughput and four FPU cycle latency. The FPU implements the multiply-add (MADD) and multiply-sub (MSUB) instructions with intermediate rounding after the multiply function. The result is guaranteed to be the same as executing a MUL and an ADD instruction separately, but the instruction latency, instruction fetch, dispatch bandwidth, and the total number of register accesses are improved.

IEEE denormalized input operands and results are supported by hardware for some instructions. IEEE denormalized results are not supported by hardware in general, but a fast flush-to-zero mode is provided to optimize performance. The fast flush-to-zero mode is enabled through the FCCR register, and use of this mode is recommended for best performance when denormalized results are generated.

The FPU has a separate pipeline for floating point instruction execution. This pipeline operates in parallel with the integer core pipeline and does not stall when the integer pipeline stalls. This allows long-running FPU operations, such as divide or square root, to be partially masked by system stalls and/or other integer unit instructions. Arithmetic instructions are always dispatched and completed in order, but loads and stores can complete out of order. The exception model is “precise” at all times.

Table 3-4 contains the floating point instruction latencies and repeat rates for the processor core. In this table, 'Latency' refers to the number of FPU cycles necessary for the first instruction to produce the result needed by the second instruction. The “Repeat Rate” refers to the maximum rate at which an instruction can be executed per FPU cycle.

TABLE 3-4: FPU INSTRUCTION LATENCIES AND REPEAT RATES

Op code	Latency (FPU Cycles)	Repeat Rate (FPU Cycles)
ABS.[S,D], NEG.[S,D], ADD.[S,D], SUB.[S,D], C.cond.[S,D], MUL.S	4	1
MADD.S, MSUB.S, NMADD.S, NMSUB.S, CABS.cond.[S,D]	4	1
CVT.D.S, CVT.PS.PW, CVT.[S,D].[W,L]	4	1
CVT.S.D, CVT.[W,L].[S,D], CEIL.[W,L].[S,D], FLOOR.[W,L].[S,D], ROUND.[W,L].[S,D], TRUNC.[W,L].[S,D]	4	1
MOV.[S,D], MOVF.[S,D], MOVN.[S,D], MOVT.[S,D], MOVZ.[S,D]	4	1
MUL.D	5	2
MADD.D, MSUB.D, NMADD.D, NMSUB.D	5	2
RECIP.S	13	10
RECIP.D	26	21
RSQRT.S	17	14
RSQRT.D	36	31
DIV.S, SQRT.S	17	14
DIV.D, SQRT.D	32	29
MTC1, DMTC1, LWC1, LDC1, LDXC1, LUXC1, LWXC1	4	1
MFC1, DMFC1, SWC1, SDC1, SDXC1, SUXC1, SWXC1	1	1

Legend: S = Single (32-bit) D = Double (64-bit) W = Word (32-bit) L = Long word (64-bit)

REGISTER 4-8: SBTxREGy: SYSTEM BUS TARGET 'x' REGION 'y' REGISTER ('x' = 0-13; 'y' = 0-8)

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
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	BASE<21:14>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	BASE<13:6>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	U-0
	BASE<5:0>						PRI	—
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
	SIZE<4:0>					—	—	—

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

bit 31-10 **BASE<21:0>:** Region Base Address bits

bit 9 **PRI:** Region Priority Level bit
1 = Level 2
0 = Level 1

bit 8 **Unimplemented:** Read as '0'

bit 7-3 **SIZE<4:0>:** Region Size bits
Permissions for a region are only active is the SIZE is non-zero.
11111 = Region size = $2^{(SIZE - 1)}$ x 1024 (bytes)
•
•
•
00001 = Region size = $2^{(SIZE - 1)}$ x 1024 (bytes)
00000 = Region is not present

bit 2-0 **Unimplemented:** Read as '0'

Note 1: Refer to Table 4-6 for the list of available targets and their descriptions.
2: For some target regions, certain bits in this register are read-only with preset values. See Table 4-6 for more information.

TABLE 7-3: INTERRUPT REGISTER MAP (CONTINUED)

Virtual Address (BF81_#)	Register Name(1)	Bit Range	Bits																All Resets
			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	
03D0	IPC41	31:16	—	—	—	FCEIP<2:0>			FCEIS<1:0>		—	—	—	RTCCIP<2:0>			RTCCIS<1:0>		0000
		15:0	—	—	—	SPI4TXIP<2:0>			SPI4TXIS<1:0>		—	—	—	SPI4RXIP<2:0>			SPI4RXIS<1:0>		0000
03E0	IPC42	31:16	—	—	—	U4RXIP<2:0>			U4RXIS<1:0>		—	—	—	U4EIP<2:0>			U4EIS<1:0>		0000
		15:0	—	—	—	SQI1IP<2:0>			SQI1IS<1:0>		—	—	—	PREIP<2:0>			PREIS<1:0>		0000
03F0	IPC43	31:16	—	—	—	I2C4MIP<2:0>			I2C4MIS<1:0>		—	—	—	I2C4SIP<2:0>			I2C4SIS<1:0>		0000
		15:0	—	—	—	I2C4BIP<2:0>			I2C4BIS<1:0>		—	—	—	U4TXIP<2:0>			U4TXIS<1:0>		0000
0400	IPC44	31:16	—	—	—	U5EIP<2:0>			U5EIS<1:0>		—	—	—	SPI5TXIP<2:0>(2)			SPI5TXIS<1:0>(2)		0000
		15:0	—	—	—	SPI5RXIP<2:0>(2)			SPI5RXIS<1:0>(2)		—	—	—	SPI5EIP<2:0>(2)			SPI5EIS<1:0>(2)		0000
0410	IPC45	31:16	—	—	—	I2C5SIP<2:0>			I2C5SIS<1:0>		—	—	—	I2C5BIP<2:0>			I2C5BIS<1:0>		0000
		15:0	—	—	—	U5TXIP<2:0>			U5TXIS<1:0>		—	—	—	U5RXIP<2:0>			U5RXIS<1:0>		0000
0420	IPC46	31:16	—	—	—	SPI6TXIP<2:0>(2)			SPI6TXIS<1:0>(2)		—	—	—	SPI6RXIP<2:0>(2)			SPI6RXIS<1:0>(2)		0000
		15:0	—	—	—	SPI6EIP<2:0>(2)			SPI6EIS<1:0>(2)		—	—	—	I2C5MIP<2:0>			I2C5MIS<1:0>		0000
0430	IPC47	31:16	—	—	—	—	—	—	—	—	—	—	—	U6TXIP<2:0>			U6TXIS<1:0>		0000
		15:0	—	—	—	U6RXIP<2:0>			U6RXIS<1:0>		—	—	—	U6EIP<2:0>			U6EIS<1:0>		0000
0440	IPC48	31:16	—	—	—	—	—	—	—	—	—	—	—	ADCURDYIP<2:0>			ADCURDYIS<1:0>		0000
		15:0	—	—	—	ADCARDYIP<2:0>			ADCARDYIS<1:0>		—	—	—	ADCEOSIP<2:0>			ADCEOSIS<1:0>		0000
0450	IPC49	31:16	—	—	—	ADC1EIP<2:0>			ADC1EIS<1:0>					ADC0EIP<2:0>			ADC0EIS<1:0>		0000
		15:0	—	—	—	—	—	—	—	—				ADCGRPIP<2:0>			ADCGRPIS<1:0>		0000
0460	IPC50	31:16	—	—	—	—	—	—	—	—				ADC4EIP<2:0>			ADC4EIS<1:0>		0000
		15:0	—	—	—	ADC3EIP<2:0>			ADC3EIS<1:0>					ADC2EIP<2:0>			ADC2EIS<1:0>		0000
0470	IPC51	31:16	—	—	—	ADC1WIP<2:0>			ADC1WIS<1:0>					ADC0WIP<2:0>			ADC0WIS<1:0>		0000
		15:0	—	—	—	ADC7EIP<2:0>			ADC7EIS<1:0>					—	—	—	—	—	0000
0480	IPC52	31:16	—	—	—	—	—	—	—	—				ADC4WIP<2:0>			ADC4WIS<1:0>		0000
		15:0	—	—	—	ADC3WIP<2:0>			ADC3WIS<1:0>					ADC2WIP<2:0>			ADC2WIS<1:0>		0000
0490	IPC53	31:16	—	—	—	—	—	—	—	—				—	—	—	—	—	0000
		15:0	—	—	—	ADC7WIP<2:0>			ADC7WIS<1:0>					—	—	—	—	—	0000
0540	OFF000	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VOFF<17:16>		0000
		15:0	VOFF<15:1>															—	0000
0544	OFF001	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VOFF<17:16>		0000
		15:0	VOFF<15:1>															—	0000

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

Note 1: 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 Registers"** for more information.
2: This bit or register is not available on 64-pin devices.
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.
8: This bit or register is not available on 124-pin devices.

REGISTER 9-1: PRECON: PREFETCH MODULE 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
31:24	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0
	—	—	—	—	—	PFMSECEN	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
	—	—	PREFEN<1:0>		—	PFMWS<2:0>(1)		

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 31-27 **Unimplemented:** Read as '0'

bit 26 **PFMSECEN:** Flash SEC Interrupt Enable bit
1 = Generate an interrupt when the PFMSEC bit (PRESTAT<26>) is set
0 = Do not generate an interrupt when the PFMSEC bit is set

bit 25-6 **Unimplemented:** Read as '0'

bit 5-4 **PREFEN<1:0>:** Predictive Prefetch Enable bits
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

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **PFMWS<2:0>:** PFM Access Time Defined in Terms of SYSCLK Wait States bits(1)
111 = Seven Wait states
•
•
•
010 = Two Wait states
001 = One Wait state
000 = Zero Wait states

Note 1: For the Wait states to SYSCLK relationship, refer to Table 37-13 in **Section37.0 "Electrical Characteristics"**.

REGISTER 10-12: DCHxSSIZ: DMA CHANNEL x SOURCE SIZE 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
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSIZ<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
	CHSSIZ<7:0>							

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 31-16 **Unimplemented:** Read as '0'

bit 15-0 **CHSSIZ<15:0>:** Channel Source Size bits

1111111111111111 = 65,535 byte source size

•
•
•

0000000000000010 = 2 byte source size

0000000000000001 = 1 byte source size

0000000000000000 = 65,536 byte source size

REGISTER 10-13: DCHxDSIZ: DMA CHANNEL x DESTINATION SIZE 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
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSIZ<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
	CHDSIZ<7:0>							

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 31-16 **Unimplemented:** Read as '0'

bit 15-0 **CHDSIZ<15:0>:** Channel Destination Size bits

1111111111111111 = 65,535 byte destination size

•
•
•

0000000000000010 = 2 byte destination size

0000000000000001 = 1 byte destination size

0000000000000000 = 65,536 byte destination size

REGISTER 11-8: USBIENCSR0: USB INDEXED ENDPOINT CONTROL STATUS REGISTER 0 (ENDPOINT 1-7) (CONTINUED)

bit 23 **INCOMPTX:** Incomplete TX Status bit (Device mode)

1 = For high-bandwidth Isochronous endpoint, a large packet has been split into 2 or 3 packets for transmission but insufficient IN tokens have been received to send all the parts

0 = Normal operation

In anything other than isochronous transfers, this bit will always return 0.

NAKTMOUT: NAK Time-out status bit (Host mode)

1 = TX endpoint is halted following the receipt of NAK responses for longer than the NAKLIM setting

0 = Written by software to clear this bit

bit 22 **CLRDT:** Clear Data Toggle Control bit

1 = Resets the endpoint data toggle to 0

0 = Do not clear the data toggle

bit 21 **SENTSTALL:** STALL handshake transmission status bit (Device mode)

1 = STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared.

0 = Written by software to clear this bit

RXSTALL: STALL receipt bit (Host mode)

1 = STALL handshake is received. Any DMA request in progress is stopped, the FIFO is completely flushed and the TXPKTRDY bit is cleared.

0 = Written by software to clear this bit

bit 20 **SENDSTALL:** STALL handshake transmission control bit (Device mode)

1 = Issue a STALL handshake to an IN token

0 = Terminate stall condition

This bit has no effect when the endpoint is being used for Isochronous transfers.

SETUPPKT: Definition bit (Host mode)

1 = When set at the same time as the TXPKTRDY bit is set, send a SETUP token instead of an OUT token for the transaction. This also clears the Data Toggle.

0 = Normal OUT token for the transaction

bit 19 **FLUSH:** FIFO Flush control bit

1 = Flush the latest packet from the endpoint TX FIFO. The FIFO pointer is reset, TXPKTRDY is cleared and an interrupt is generated.

0 = Do not flush the FIFO

bit 18 **UNDERRUN:** Underrun status bit (Device mode)

1 = An IN token has been received when TXPKTRDY is not set.

0 = Written by software to clear this bit.

ERROR: Handshake failure status bit (Host mode)

1 = Three attempts have been made to send a packet and no handshake packet has been received

0 = Written by software to clear this bit.

bit 17 **FIFONE:** FIFO Not Empty status bit

1 = There is at least 1 packet in the TX FIFO

0 = TX FIFO is empty

bit 16 **TXPKTRDY:** TX Packet Ready Control bit

The software sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. This bit is also automatically cleared prior to loading a second packet into a double-buffered FIFO.

REGISTER 11-9: USBIENCSR1: USB INDEXED ENDPOINT CONTROL STATUS REGISTER 1 (ENDPOINT 1-7) (CONTINUED)

bit 18 **OVERRUN:** Data Overrun Status bit (*Device mode*)

1 = An OUT packet cannot be loaded into the RX FIFO.
0 = Written by software to clear this bit

This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero.

ERROR: No Data Packet Received Status bit (*Host mode*)

1 = Three attempts have been made to receive a packet and no data packet has been received. An interrupt is generated.

0 = Written by the software to clear this bit.

This bit is only valid when the RX endpoint is operating in Bulk or Interrupt mode. In ISO mode, it always returns zero.

bit 17 **FIFOFULL:** FIFO Full Status bit

1 = No more packets can be loaded into the RX FIFO
0 = The RX FIFO has at least one free space

bit 16 **RXPKTRDY:** Data Packet Reception Status bit

1 = A data packet has been received. An interrupt is generated.
0 = Written by software to clear this bit when the packet has been unloaded from the RX FIFO.

bit 15-11 **MULT<4:0>:** Multiplier Control bits

For Isochronous/Interrupt endpoints or of packet splitting on Bulk endpoints, multiplies TXMAXP by MULT+1 for the payload size.

For Bulk endpoints, MULT can be up to 32 and defines the number of “USB” packets of the specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. The data packet is required to be an exact multiple of the payload specified by TXMAXP.

For Isochronous/Interrupts endpoints operating in Hi-Speed mode, MULT may be either 2 or 3 and specifies the maximum number of such transactions that can take place in a single microframe.

bit 10-0 **RXMAXP<10:0>:** Maximum RX Payload Per Transaction Control bits

This field sets the maximum payload (in bytes) transmitted in a single transaction. The value is subject to the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transfers in Full-Speed and Hi-Speed operations.

RXMAXP must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

TABLE 12-23: PERIPHERAL PIN SELECT OUTPUT REGISTER MAP (CONTINUED)

Virtual Address (BF80_#)	Register Name	Bit Range	Bits															All Resets
			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	
15B4	RPC13R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPC13R<3:0>				0000
15B8	RPC14R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPC14R<3:0>				0000
15C0	RPD0R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD0R<3:0>				0000
15C4	RPD1R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD1R<3:0>				0000
15C8	RPD2R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD2R<3:0>				0000
15CC	RPD3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD3R<3:0>				0000
15D0	RPD4R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD4R<3:0>				0000
15D4	RPD5R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD5R<3:0>				0000
15D8	RPD6R(2)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD6R<3:0>				0000
15DC	RPD7R(2)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD7R<3:0>				0000
15E4	RPD9R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD9R<3:0>				0000
15E8	RPD10R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD10R<3:0>				0000
15EC	RPD11R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD11R<3:0>				0000
15F0	RPD12R(1)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD12R<3:0>				0000
15F8	RPD14R(1)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD14R<3:0>				0000
15FC	RPD15R(1)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPD15R<3:0>				0000
160C	RPE3R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPE3R<3:0>				0000
1614	RPE5R	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	RPE5R<3:0>				0000

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

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

2: This register is not available on 64-pin and 100-pin devices.

REGISTER 14-1: TxCON: TYPE B TIMER CONTROL REGISTER ('x' = 2-9) (CONTINUED)

bit 2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timer Clock Source Select bit(1)
1 = External clock from TxCK pin
0 = Internal peripheral clock

bit 0 **Unimplemented:** Read as '0'

Note 1: While operating in 32-bit mode, this bit has no effect for odd numbered timers (Timer1, Timer3, Timer5, Timer7, and Timer9). All timer functions are set through the even numbered timers.

2: While operating in 32-bit mode, this bit must be cleared on odd numbered timers to enable the 32-bit timer in Idle mode.

3: This bit is available only on even numbered timers (Timer2, Timer4, Timer6, and Timer8).

REGISTER 29-22: CiFIFOUAn: CAN FIFO USER ADDRESS REGISTER 'n' ('n' = 0-31)

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
31:24	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUAn<31:24>							
23:16	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUAn<23:16>							
15:8	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUAn<15:8>							
7:0	R-x	R-x	R-x	R-x	R-x	R-x	R-0(1)	R-0(1)
	CiFIFOUAn<7:0>							

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 **CiFIFOUAn<31:0>:** CAN FIFO User Address bits

TXEN = 1: (FIFO configured as a Transmit Buffer)

A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0: (FIFO configured as a Receive Buffer)

A read of this register will return the address where the next message is to be read (FIFO tail).

Note 1: This bit will always read '0', which forces byte-alignment of messages.

Note: This register is not guaranteed to read correctly in Configuration mode, and should only be accessed when the module is not in Configuration mode.

REGISTER 29-23: CiFIFOCIn: CAN MODULE MESSAGE INDEX REGISTER 'n' ('n' = 0-31)

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
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
	—	—	—	CiFIFOCIn<4:0>				

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 31-5 **Unimplemented:** Read as '0'

bit 4-0 **CiFIFOCIn<4:0>:** CAN Side FIFO Message Index bits

TXEN = 1: (FIFO configured as a Transmit Buffer)

A read of this register will return an index to the message that the FIFO will next attempt to transmit.

TXEN = 0: (FIFO configured as a Receive Buffer)

A read of this register will return an index to the message that the FIFO will use to save the next message.

REGISTER 30-14: ETHIRQ: ETHERNET CONTROLLER INTERRUPT REQUEST REGISTER

bit 7 **RXDONE:** Receive Done Interrupt bit(2)

1 = RX packet was successfully received
0 = No interrupt pending

This bit is set whenever an RX packet is successfully received. It is cleared by either a Reset or CPU write of a '1' to the CLR register.

bit 6 **PKTPEND:** Packet Pending Interrupt bit(2)

1 = RX packet pending in memory
0 = RX packet is not pending in memory

This bit is set when the BUFCNT counter has a value other than '0'. It is cleared by either a Reset or by writing the BUFCDEC bit to decrement the BUFCNT counter. Writing a '0' or a '1' has no effect.

bit 5 **RXACT:** Receive Activity Interrupt bit(2)

1 = RX packet data was successfully received
0 = No interrupt pending

This bit is set whenever RX packet data is stored in the RXBM FIFO. It is cleared by either a Reset or CPU write of a '1' to the CLR register.

bit 4 **Unimplemented:** Read as '0'

bit 3 **TXDONE:** Transmit Done Interrupt bit(2)

1 = TX packet was successfully sent
0 = No interrupt pending

This bit is set when the currently transmitted TX packet completes transmission, and the Transmit Status Vector is loaded into the first descriptor used for the packet. It is cleared by either a Reset or CPU write of a '1' to the CLR register.

bit 2 **TXABORT:** Transmit Abort Condition Interrupt bit(2)

1 = TX abort condition occurred on the last TX packet
0 = No interrupt pending

This bit is set when the MAC aborts the transmission of a TX packet for one of the following reasons:

- Jumbo TX packet abort
- Underrun abort
- Excessive defer abort
- Late collision abort
- Excessive collisions abort

This bit is cleared by either a Reset or CPU write of a '1' to the CLR register.

bit 1 **RXBUFNA:** Receive Buffer Not Available Interrupt bit(2)

1 = RX Buffer Descriptor Not Available condition has occurred
0 = No interrupt pending

This bit is set by a RX Buffer Descriptor Overrun condition. It is cleared by either a Reset or a CPU write of a '1' to the CLR register.

bit 0 **RXOVFLW:** Receive FIFO Over Flow Error bit(2)

1 = RX FIFO Overflow Error condition has occurred
0 = No interrupt pending

RXOVFLW is set by the RXBM Logic for an RX FIFO Overflow condition. It is cleared by either a Reset or CPU write of a '1' to the CLR register.

Note 1: This bit is only used for TX operations.
2: This bit is are only used for RX operations.

Note:	It is recommended to use the SET, CLR, or INV registers to set or clear any bit in this register. Setting or clearing any bits in this register should only be done for debug/test purposes.

REGISTER 30-19: ETHMCOLFRM: ETHERNET CONTROLLER MULTIPLE COLLISION FRAMES STATISTICS 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
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	MCOLFRMCNT<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
	MCOLFRMCNT<7:0>							

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 31-16 **Unimplemented:** Read as '0'

bit 15-0 **MCOLFRMCNT<15:0>:** Multiple Collision Frame Count bits

Increment count for frames that were successfully transmitted after there was more than one collision.

Note 1: This register is only used for TX operations.

2: This register is automatically cleared by hardware after a read operation, unless the byte enables for bytes 0/1 are '0'.

3: It is recommended to use the SET, CLR, or INV registers to set or clear any bit in this register. Setting or clearing any bits in this register should only be done for debug/test purposes.

REGISTER 34-5: DEVCFG2/ADEVCFG2: DEVICE CONFIGURATION WORD 2 (CONTINUED)

bit 3 **Reserved:** Write as '1'

bit 2-0 **FPLLIDIV<2:0>:** PLL Input Divider bits

111 = Divide by 8
110 = Divide by 7
101 = Divide by 6
100 = Divide by 5
011 = Divide by 4
010 = Divide by 3
001 = Divide by 2
000 = Divide by 1

TABLE 37-10: DC CHARACTERISTICS: I/O PIN INPUT INJECTION CURRENT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.1V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typical(1)**	**Max.**	**Units**	**Conditions**
DI60a	IICL	**Input Low Injection Current**	0	—	-5(2,5)	mA	This parameter applies to all pins, with the exception of RB10. Maximum IICH current for this exception is 0 mA.
DI60b	IICH	**Input High Injection Current**	0	—	+5(3,4,5)	mA	This parameter applies to all pins, with the exception of all 5V tolerant pins, OSCI, OSCO, SOSCI, SOSCO, D+, D- and RB10. Maximum IICH current for these exceptions is 0 mA.
DI60c	∑IICT	**Total Input Injection Current (sum of all I/O and control pins)**	-20(6)	—	+20(6)	mA	Absolute instantaneous sum of all ± input injection currents from all I/O pins (| IICL + | IICH |) ≤ ∑IICT

Note 1: Data in “Typical” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: VIL source < (VSS - 0.3). Characterized but not tested.

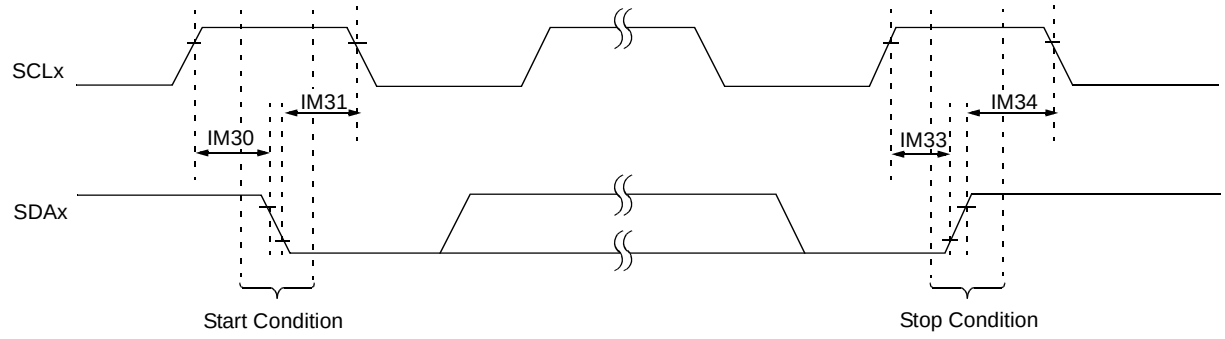
3: VIH source > (VDD + 0.3) for non-5V tolerant pins only.

4: Digital 5V tolerant pins do not have an internal high side diode to VDD, and therefore, cannot tolerate any “positive” input injection current.

5: Injection currents > | 0 | can affect the ADC results by approximately 4 to 6 counts (i.e., VIH Source > (VDD + 0.3) or VIL source < (VSS - 0.3)).

6: Any number and/or combination of I/O pins not excluded under IICL or IICH conditions are permitted provided the “absolute instantaneous” sum of the input injection currents from all pins do not exceed the specified limit. If **Note 2**, IICL = (((Vss - 0.3) - VIL source) / Rs). If **Note 3**, IICH = ((IICH source - (VDD + 0.3)) / RS). RS = Resistance between input source voltage and device pin. If (VSS - 0.3) ≤ VSOURCE ≤ (VDD + 0.3), injection current = 0.

FIGURE 37-16: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 37-1 for load conditions.

FIGURE 37-17: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

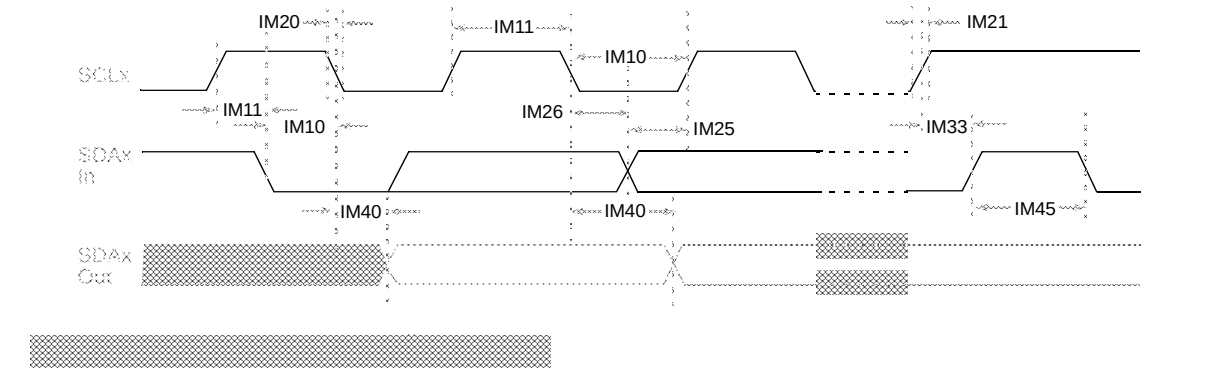


TABLE 37-35: I2Cx BUS DATA TIMING REQUIREMENTS (MASTER MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 2.1V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param. No.	**Symbol**	**Characteristics**		**Min.(1)**	**Max.**	**Units**	**Conditions**
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TPBCLK2 * (BRG + 2)	—	µs	—
			400 kHz mode	TPBCLK2 * (BRG + 2)	—	µs	—
			1 MHz mode **(Note 2)**	TPBCLK2 * (BRG + 2)	—	µs	—
IM11	THI:SCL	Clock High Time	100 kHz mode	TPBCLK2 * (BRG + 2)	—	µs	—
			400 kHz mode	TPBCLK2 * (BRG + 2)	—	µs	—
			1 MHz mode **(Note 2)**	TPBCLK2 * (BRG + 2)	—	µs	—
IM20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode **(Note 2)**	—	100	ns	

Note 1: BRG is the value of the I^2C Baud Rate Generator.

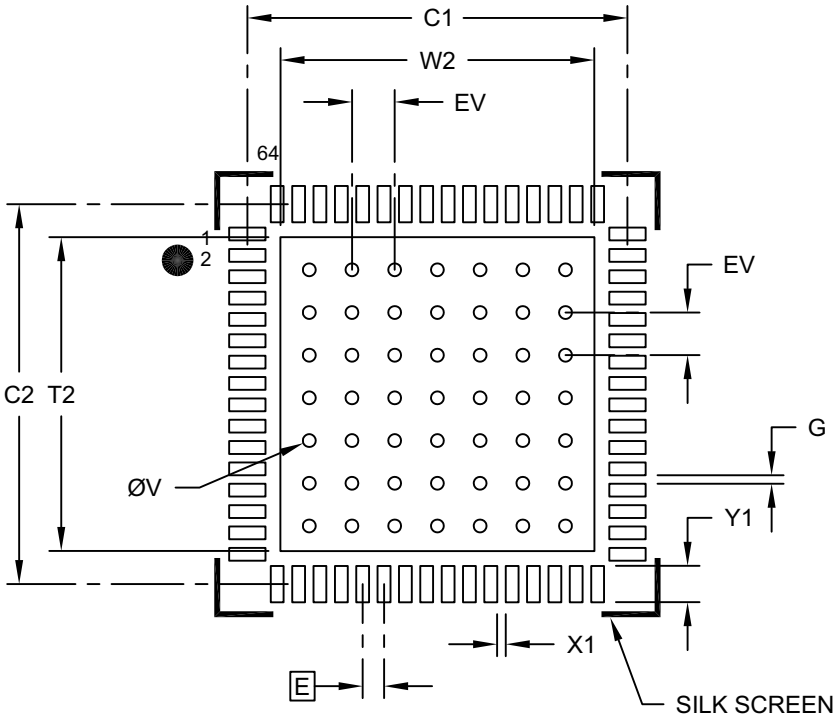
2: Maximum pin capacitance = 10 pF for all I2Cx pins (for 1 MHz mode only).

3: The typical value for this parameter is 104 ns.

41.2 Package Details

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]
With 0.40 mm Contact Length and 7.70x7.70mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			7.50
Optional Center Pad Length	T2			7.50
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X20)	X1			0.30
Contact Pad Length (X20)	Y1			0.90
Contact Pad to Center Pad (X20)	G	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing No. C04-2213B

TABLE A-1: OSCILLATOR CONFIGURATION DIFFERENCES (CONTINUED)

PIC32MX5XX/6XX/7XX Feature	PIC32MZ EF Feature
Secondary Oscillator Enable	
FSOSCEN (DEVCFG1<**5**>)	The location of the SOSCEN bit in the Flash Configuration Words has moved. FSOSCEN (DEVCFG1<**6**>)
PLL Configuration	
The FNOSC<2:0> and NOSC<2:0> bits select between POSC and FRC. **FNOSC<2:0> (DEVCFG1<2:0>)** **NOSC<2:0> (OSCCON<10:8>)**	Selection of which input clock (POSC or FRC) is now done through the FPLLICLK/PLLICLK bits. **FPLLICLK (DEVCFG2<7>)** **PLLICLK (SPLLCON<7>)**
On PIC32MX devices, the input frequency to the PLL had to be between 4 MHz and 5 MHz. FPLLIDIV selected how to divide the input frequency to give it the appropriate range. FPLLIDIV<2:0> (DEVCFG2<2:0>) **111 = 12x divider** **110 = 10x divider** 101 = 6x divider 100 = 5x divider 011 = 4x divider 010 = 3x divider 001 = 2x divider 000 = 1x divider	On PIC32MZ EF devices, the input range for the PLL is wider (5 MHz to 64 MHz). The input divider values have changed, and new FPLLRNG/PLLRNG bits have been added to indicate under what range the input frequency falls. FPLLIDIV<2:0> (DEVCFG2<2:0>) **PLLIDIV<2:0> (SPLLCON<2:0>)** **111 = Divide by 8** **110 = Divide by 7** 101 = Divide by 6 100 = Divide by 5 011 = Divide by 4 010 = Divide by 3 001 = Divide by 2 000 = Divide by 1 **FPLLRNG<2:0> (DEVCFG2<6:4>)** **PLLRNG<2:0> (SPLLCON<2:0>)** **111 = Reserved** **110 = Reserved** **101 = 34-64 MHz** **100 = 21-42 MHz** **011 = 13-26 MHz** **010 = 8-16 MHz** **001 = 5-10 MHz** **000 = Bypass**
On PIC32MX devices, the output frequency of PLL is between 60 MHz and 120 MHz. The PLL multiplier and divider bits configure the PLL for this range. FPLLMUL<**2**:0> (DEVCFG2<**6:4**>) PLLMULT<**2**:0> (OSCCON<18:16>) **111 = 24x multiplier** **110 = 21x multiplier** **101 = 20x multiplier** **100 = 19x multiplier** **011 = 18x multiplier** **010 = 17x multiplier** **001 = 16x multiplier** **000 = 15x multiplier** FPLLODIV<2:0> (DEVCFG2<18:16>) PLLODIV<2:0> (OSCCON<29:27>) **111 = 24x multiplier** **110 = 21x multiplier** **101 = 20x multiplier** **100 = 19x multiplier** **011 = 18x multiplier** **010 = 17x multiplier** **001 = 16x multiplier** **000 = 15x multiplier**	The PLL multiplier and divider on PIC32MZ EF devices have a wider range to accommodate the wider PLL specification range. FPLLMUL**T**<**6**:0> (DEVCFG2<**14:8**>) PLLMULT<**6**:0> (**SPLLCON<22:16>**) **1111111 = Multiply by 128** **1111110 = Multiply by 127** **1111101 = Multiply by 126** **1111100 = Multiply by 125** • • • **0000000 = Multiply by 1** FPLLODIV<2:0> (DEVCFG2<18:16>) PLLODIV<2:0> (**SPLLCON<26:24>**) **111 = PLL Divide by 32** **110 = PLL Divide by 32** **101 = PLL Divide by 32** **100 = PLL Divide by 16** **011 = PLL Divide by 8** **010 = PLL Divide by 4** **001 = PLL Divide by 2** **000 = PLL Divide by 2**