



Welcome to [E-XFL.COM](https://www.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® M4K™
Core Size	32-Bit Single-Core
Speed	80MHz
Connectivity	I ² C, IrDA, LINbus, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	53
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx340f256h-80i-mr

TABLE 2: PIC32MX USB – FEATURES

USB															
Device	Pins	Packages ⁽²⁾	MHz	Program Memory (KB)	Data Memory (KB)	Timers/Capture/Compare	Programmable DMA Channels	Dedicated USB DMA Channels	VREG	Trace	EUART/SPI/I ² C™	10-bit ADC (ch)	Comparators	PMP/PSP	JTAG
PIC32MX420F032H	64	PT, MR	40	32 + 12 ⁽¹⁾	8	5/5/5	0	2	Yes	No	2/1/2	16	2	Yes	Yes
PIC32MX440F128H	64	PT, MR	80	128 + 12 ⁽¹⁾	32	5/5/5	4	2	Yes	No	2/1/2	16	2	Yes	Yes
PIC32MX440F256H	64	PT, MR	80	256 + 12 ⁽¹⁾	32	5/5/5	4	2	Yes	No	2/1/2	16	2	Yes	Yes
PIC32MX440F512H	64	PT, MR	80	512 + 12 ⁽¹⁾	32	5/5/5	4	2	Yes	No	2/1/2	16	2	Yes	Yes
PIC32MX440F128L	100	PT	80	128 + 12 ⁽¹⁾	32	5/5/5	4	2	Yes	No	2/2/2	16	2	Yes	Yes
	121	BG													
PIC32MX460F256L	100	PT	80	256 + 12 ⁽¹⁾	32	5/5/5	4	2	Yes	Yes	2/2/2	16	2	Yes	Yes
	121	BG													
PIC32MX460F512L	100	PT	80	512 + 12 ⁽¹⁾	32	5/5/5	4	2	Yes	Yes	2/2/2	16	2	Yes	Yes
	121	BG													

Legend: PT = TQFP MR = QFN BG = XBGA

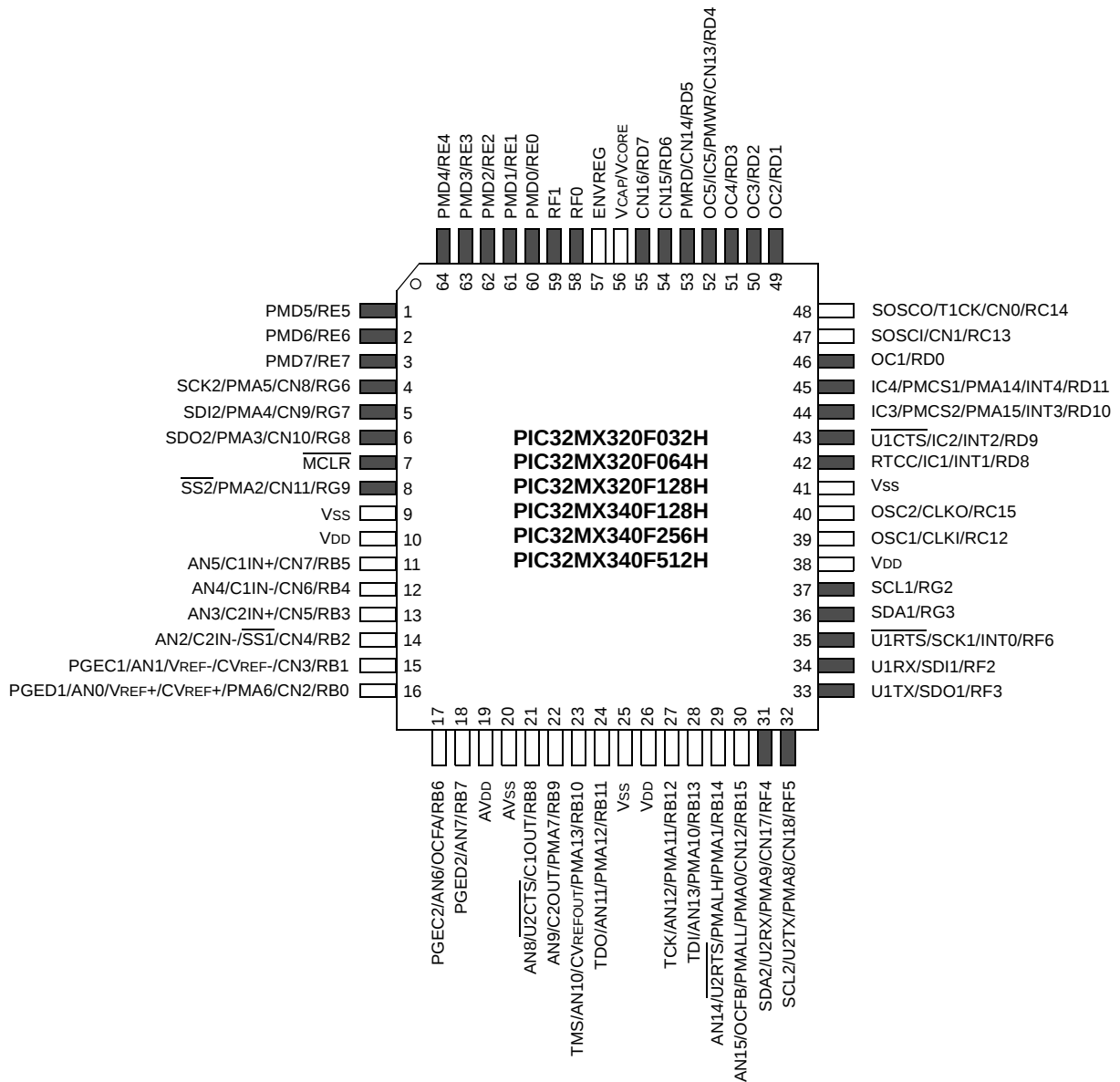
Note 1: This device features 12 KB Boot Flash memory.

2: See Legend for an explanation of the acronyms. See **Section 30.0 “Packaging Information”** for details.

Pin Diagrams (Continued)

64-Pin TQFP (General Purpose)

■ = Pins are up to 5V tolerant

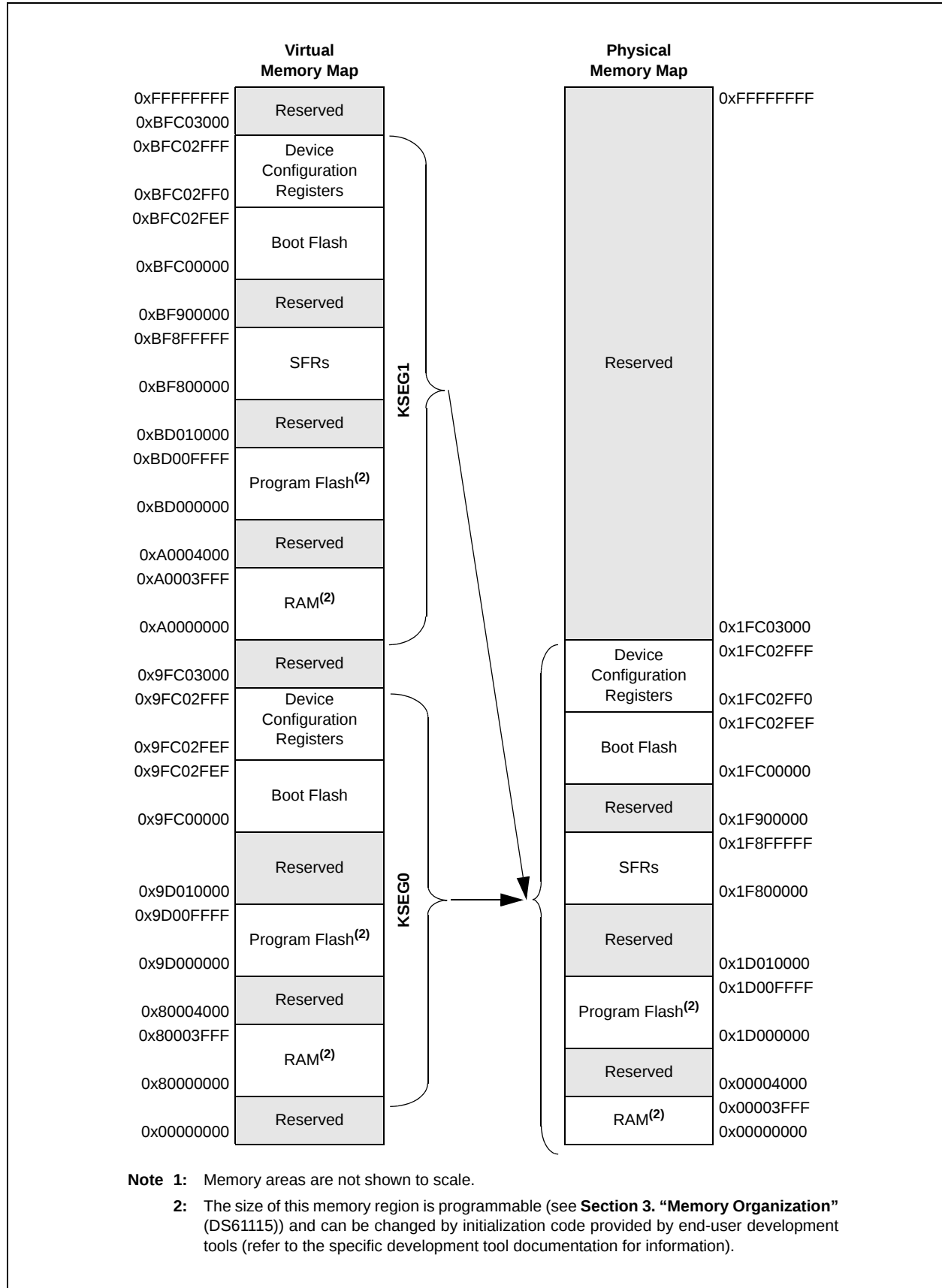


PIC32MX3XX/4XX

Table of Contents

1.0	Device Overview	21
2.0	Guidelines for Getting Started with 32-bit Microcontrollers	31
3.0	CPU	37
4.0	Memory Organization	43
5.0	Flash Program Memory	85
6.0	Resets	87
7.0	Interrupt Controller	89
8.0	Oscillator Configuration	93
9.0	Prefetch Cache	95
10.0	Direct Memory Access (DMA) Controller	97
11.0	USB On-The-Go (OTG)	99
12.0	I/O Ports	101
13.0	Timer1	103
14.0	Timer2/3 and Timer4/5	105
15.0	Input Capture	107
16.0	Output Compare	109
17.0	Serial Peripheral Interface (SPI)	111
18.0	Inter-Integrated Circuit™ (I ² C™)	113
19.0	Universal Asynchronous Receiver Transmitter (UART)	115
20.0	Parallel Master Port (PMP)	119
21.0	Real-Time Clock and Calendar (RTCC)	121
22.0	10-bit Analog-to-Digital Converter (ADC)	123
23.0	Comparator	125
24.0	Comparator Voltage Reference (CVREF)	127
25.0	Power-Saving Features	129
26.0	Special Features	131
27.0	Instruction Set	141
28.0	Development Support	147
29.0	Electrical Characteristics	151
30.0	Packaging Information	191
	Index	209

FIGURE 4-2: MEMORY MAP ON RESET FOR PIC32MX320F064H DEVICE⁽¹⁾



PIC32MX3XX/4XX

FIGURE 4-3: MEMORY MAP ON RESET FOR PIC32MX320F128H AND PIC32MX320F128L DEVICES⁽¹⁾

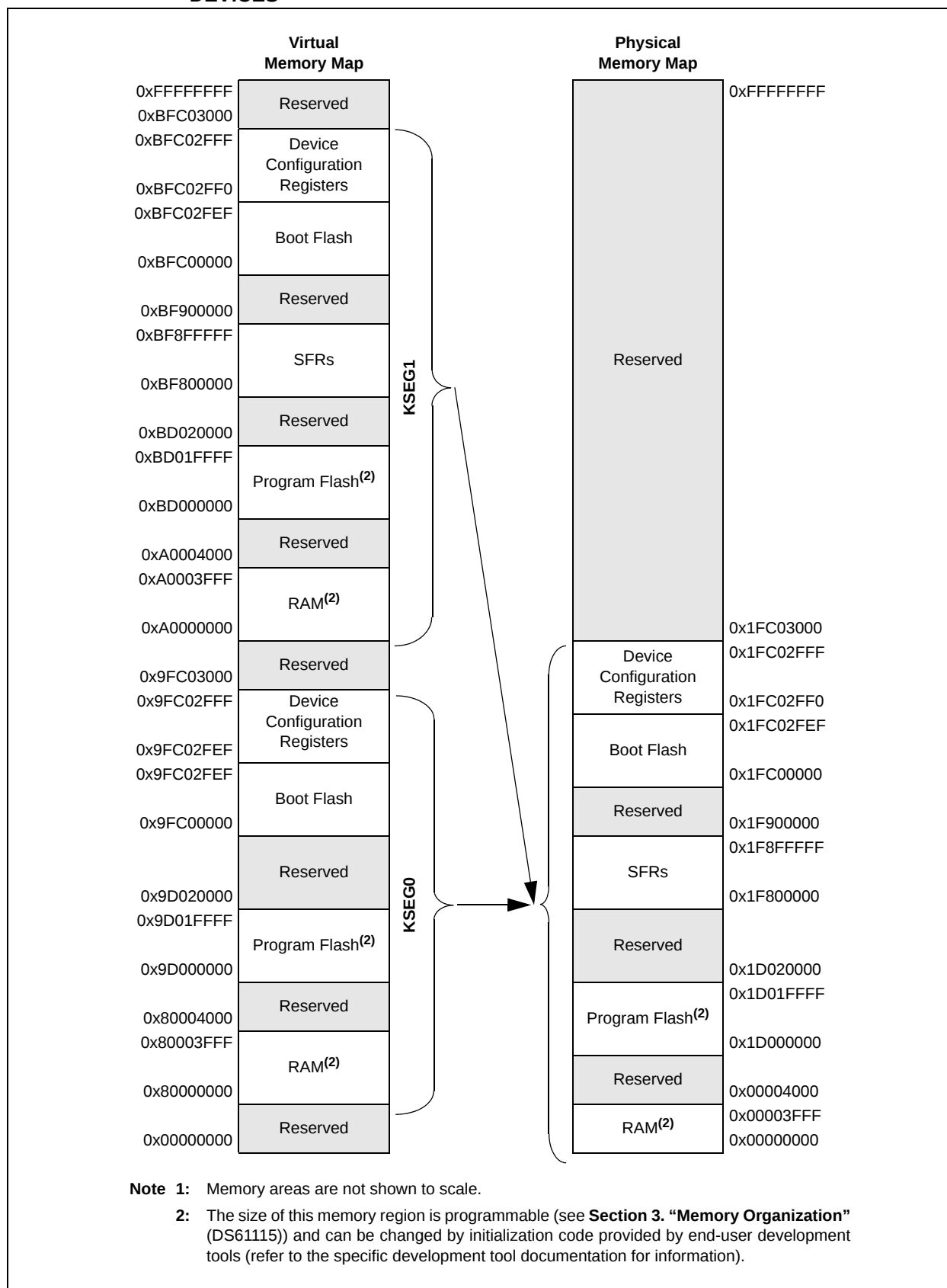


TABLE 4-29: PORTF REGISTERS MAP FOR PIC32MX320F128L, PIC32MX340F128L, PIC32MX360F256L AND PIC32MX360F512L DEVICES ONLY⁽¹⁾

Virtual Address (BF88_#)	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	
6140	TRISF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	TRISF13	TRISF12	—	—	—	TRISF8	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	31FF
6150	PORTF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	RF13	RF12	—	—	—	RF8	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
6160	LATF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	LATF13	LATF12	—	—	—	LATF8	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
6170	ODCF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	ODCF13	ODCF12	—	—	—	ODCF8	ODCF7	ODCF6	ODCF5	ODCF4	ODCF3	ODCF2	ODCF1	ODCF0	0000

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

Note 1: All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See **Section 12.1.1 “CLR, SET and INV Registers”** for more information.

TABLE 4-30: PORTF REGISTERS MAP FOR PIC32MX440F128L, PIC32MX460F256L AND PIC32MX460F512L DEVICES ONLY⁽¹⁾

Virtual Address (BF88_#)	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	
6140	TRISF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	TRISF13	TRISF12	—	—	—	TRISF8	—	—	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	313F
6150	PORTF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	RF13	RF12	—	—	—	RF8	—	—	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
6160	LATF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	LATF13	LATF12	—	—	—	LATF8	—	—	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
6170	ODCF	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	ODCF13	ODCF12	—	—	—	ODCF8	—	—	ODCF5	ODCF4	ODCF3	ODCF2	ODCF1	ODCF0	0000

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

Note 1: All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See **Section 12.1.1 “CLR, SET and INV Registers”** for more information.

10.0 DIRECT MEMORY ACCESS (DMA) CONTROLLER

Note 1: This data sheet summarizes the features of the PIC32MX3XX/4XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 31. “Direct Memory Access (DMA) Controller”** (DS61117) of the “PIC32 Family Reference Manual”, which is available from the Microchip web site (www.microchip.com/PIC32).

2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 “Memory Organization”** in this data sheet for device-specific register and bit information.

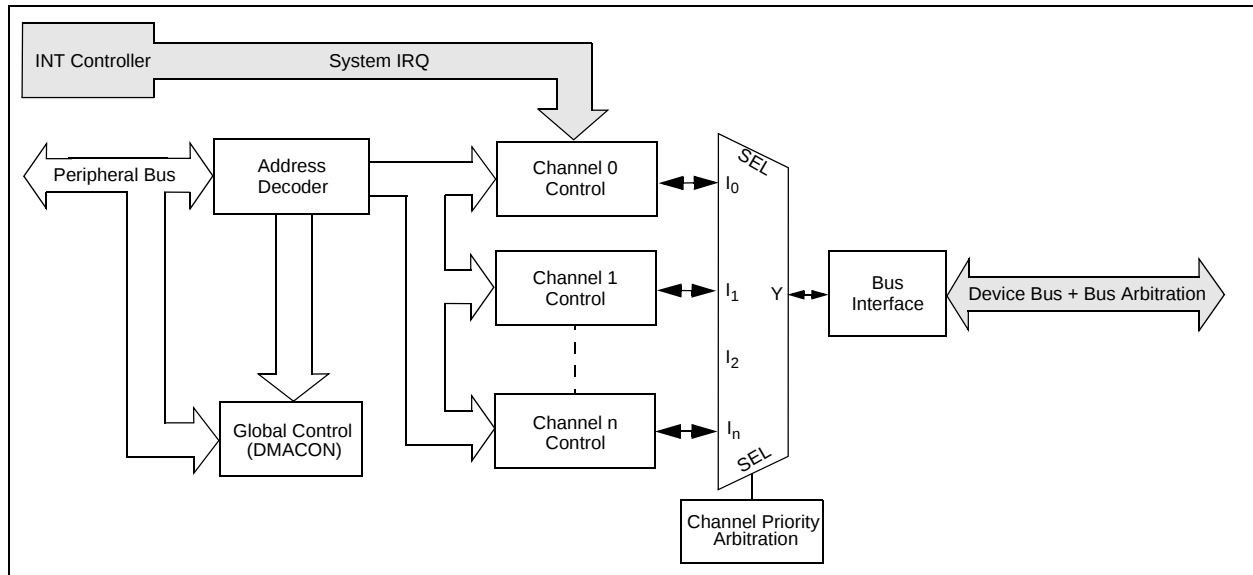
The PIC32MX Direct Memory Access (DMA) controller is a bus master module useful for data transfers between different devices without CPU intervention. The source and destination of a DMA transfer can be any of the memory mapped modules existent in the PIC32MX (such as Peripheral Bus (PBUS) devices: SPI, UART, PMP, and so on) or memory itself.

Following are some of the key features of the DMA controller module:

- Four Identical Channels, each featuring:
 - Auto-Increment Source and Destination Address Registers
 - Source and Destination Pointers
 - Memory to Memory and Memory to Peripheral Transfers

- Automatic Word-Size Detection:
 - Transfer Granularity, down to byte level
 - Bytes need not be word-aligned at source and destination
- Fixed Priority Channel Arbitration
- Flexible DMA Channel Operating Modes:
 - Manual (software) or automatic (interrupt) DMA requests
 - One-Shot or Auto-Repeat Block Transfer modes
 - Channel-to-channel chaining
- Flexible DMA Requests:
 - A DMA request can be selected from any of the peripheral interrupt sources
 - Each channel can select any (appropriate) observable interrupt as its DMA request source
 - A DMA transfer abort can be selected from any of the peripheral interrupt sources
 - Pattern (data) match transfer termination
- Multiple DMA Channel Status Interrupts:
 - DMA channel block transfer complete
 - Source empty or half empty
 - Destination full or half-full
 - DMA transfer aborted due to an external event
 - Invalid DMA address generated
- DMA Debug Support Features:
 - Most recent address accessed by a DMA channel
 - Most recent DMA channel to transfer data
- CRC Generation Module:
 - CRC module can be assigned to any of the available channels
 - CRC module is highly configurable

FIGURE 10-1: DMA BLOCK DIAGRAM



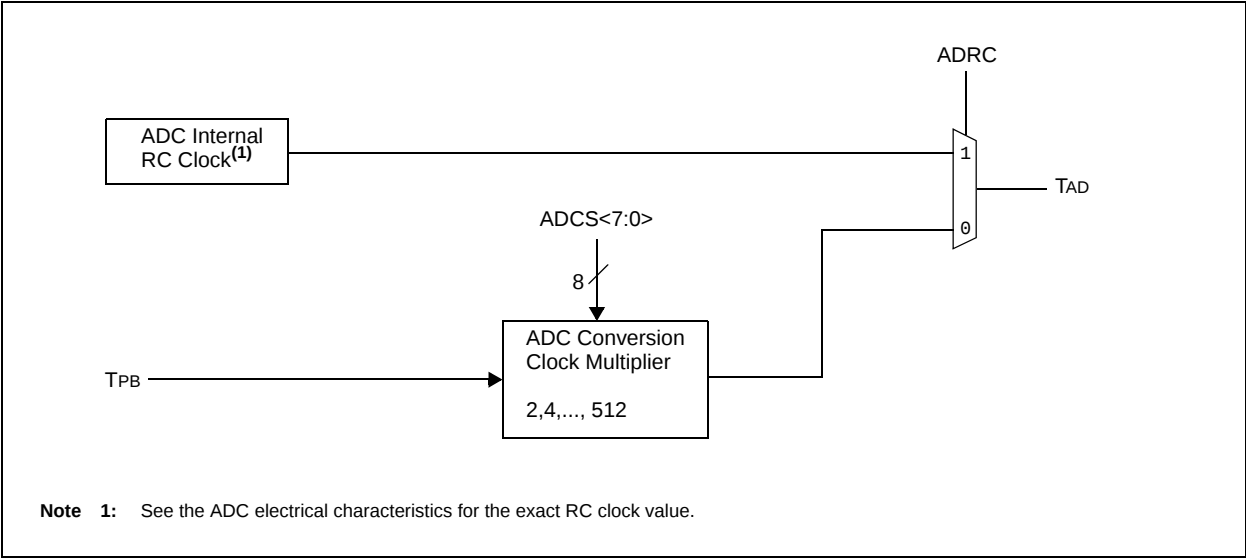
PIC32MX3XX/4XX

NOTES:

PIC32MX3XX/4XX

NOTES:

FIGURE 22-2: ADC CONVERSION CLOCK PERIOD BLOCK DIAGRAM



PIC32MX3XX/4XX

NOTES:

26.2 Watchdog Timer (WDT)

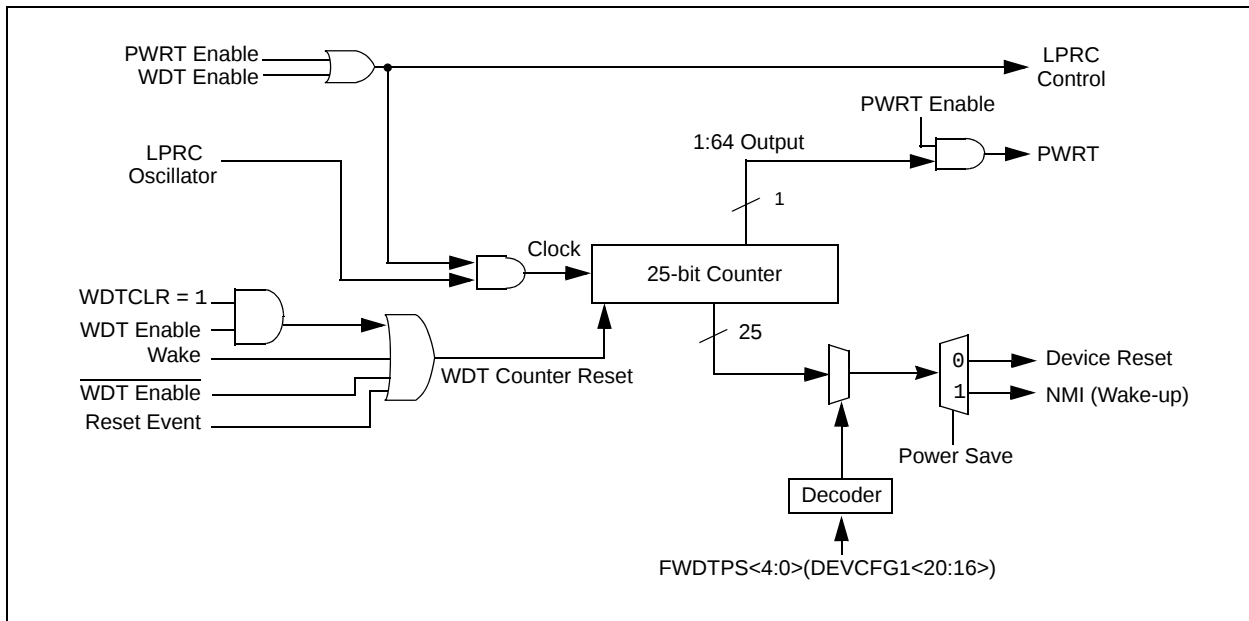
This section describes the operation of the WDT and Power-Up Timer of the PIC32MX3XX/4XX.

The WDT, when enabled, operates from the internal Low-Power Oscillator (LPRC) clock source and can be used to detect system software malfunctions by resetting the device if the WDT is not cleared periodically in software. Various WDT time-out periods can be selected using the WDT postscaler. The WDT can also be used to wake the device from Sleep or Idle mode.

The following are some of the key features of the WDT module:

- Configuration or software controlled
- User-configurable time-out period
- Can wake the device from Sleep or Idle

FIGURE 26-1: WATCHDOG AND POWER-UP TIMER BLOCK DIAGRAM



27.0 INSTRUCTION SET

The PIC32MX3XX/4XX family instruction set complies with the MIPS32 Release 2 instruction set architecture. PIC32MX does not support the following features:

- CoreExtend instructions
- Coprocessor 1 instructions
- Coprocessor 2 instructions

Table 27-1 provides a summary of the instructions that are implemented by the PIC32MX3XX/4XX family core.

Note: Refer to “MIPS32® Architecture for Programmers Volume II: The MIPS32® Instruction Set” at www.mips.com for more information.

TABLE 27-1: MIPS32® INSTRUCTION SET

Instruction	Description	Function
ADD	Integer Add	$Rd = Rs + Rt$
ADDI	Integer Add Immediate	$Rt = Rs + \text{Immed}$
ADDIU	Unsigned Integer Add Immediate	$Rt = Rs +_U \text{Immed}$
ADDU	Unsigned Integer Add	$Rd = Rs +_U Rt$
AND	Logical AND	$Rd = Rs \& Rt$
ANDI	Logical AND Immediate	$Rt = Rs \& (0_{16} \parallel \text{Immed})$
B	Unconditional Branch (Assembler idiom for: BEQ r0, r0, offset)	$PC += (\text{int})\text{offset}$
BAL	Branch and Link (Assembler idiom for: BGEZAL r0, offset)	$GPR[31] = PC + 8$ $PC += (\text{int})\text{offset}$
BEQ	Branch on Equal	if $Rs == Rt$ $PC += (\text{int})\text{offset}$
BEQL	Branch on Equal Likely ⁽¹⁾	if $Rs == Rt$ $PC += (\text{int})\text{offset}$ else Ignore Next Instruction
BGEZ	Branch on Greater Than or Equal to Zero	if $!Rs[31]$ $PC += (\text{int})\text{offset}$
BGEZAL	Branch on Greater Than or Equal to Zero and Link	$GPR[31] = PC + 8$ if $!Rs[31]$ $PC += (\text{int})\text{offset}$
BGEZALL	Branch on Greater Than or Equal to Zero and Link Likely ⁽¹⁾	$GPR[31] = PC + 8$ if $!Rs[31]$ $PC += (\text{int})\text{offset}$ else Ignore Next Instruction
BGEZL	Branch on Greater Than or Equal to Zero Likely ⁽¹⁾	if $!Rs[31]$ $PC += (\text{int})\text{offset}$ else Ignore Next Instruction
BGTZ	Branch on Greater Than Zero	if $!Rs[31] \&\& Rs != 0$ $PC += (\text{int})\text{offset}$
BGTZL	Branch on Greater Than Zero Likely ⁽¹⁾	if $!Rs[31] \&\& Rs != 0$ $PC += (\text{int})\text{offset}$ else Ignore Next Instruction
BLEZ	Branch on Less Than or Equal to Zero	if $Rs[31] \parallel Rs == 0$ $PC += (\text{int})\text{offset}$

Note 1: This instruction is deprecated and should not be used.

TABLE 27-1: MIPS32® INSTRUCTION SET (CONTINUED)

Instruction	Description	Function
JAL	Jump and Link	$GPR[31] = PC + 8$ $PC = PC[31:28] \parallel offset \ll 2$
JALR	Jump and Link Register	$Rd = PC + 8$ $PC = Rs$
JALR.HB	Jump and Link Register with Hazard Barrier	Like JALR, but also clears execution and instruction hazards
JR	Jump Register	$PC = Rs$
JR.HB	Jump Register with Hazard Barrier	Like JR, but also clears execution and instruction hazards
LB	Load Byte	$Rt = (\text{byte})Mem[Rs+offset]$
LBU	Unsigned Load Byte	$Rt = (\text{ubyte})Mem[Rs+offset]$
LH	Load Halfword	$Rt = (\text{half})Mem[Rs+offset]$
LHU	Unsigned Load Halfword	$Rt = (\text{uhalf})Mem[Rs+offset]$
LL	Load Linked Word	$Rt = Mem[Rs+offset]$ $LL_{bit} = 1$ $LLAdr = Rs + offset$
LUI	Load Upper Immediate	$Rt = \text{immediate} \ll 16$
LW	Load Word	$Rt = Mem[Rs+offset]$
LWPC	Load Word, PC relative	$Rt = Mem[PC+offset]$
LWL	Load Word Left	$Re = Re \text{ MERGE } Mem[Rs+offset]$
LWR	Load Word Right	$Re = Re \text{ MERGE } Mem[Rs+offset]$
MADD	Multiply-Add	$HI \mid LO += (\text{int})Rs * (\text{int})Rt$
MADDU	Multiply-Add Unsigned	$HI \mid LO += (\text{uns})Rs * (\text{uns})Rt$
MFC0	Move from Coprocessor 0	$Rt = CPR[0, Rd, sel]$
MFHI	Move from HI	$Rd = HI$
MFLO	Move from LO	$Rd = LO$
MOVN	Move Conditional on Not Zero	if $Rt \neq 0$ then $Rd = Rs$
MOVZ	Move Conditional on Zero	if $Rt = 0$ then $Rd = Rs$
MSUB	Multiply-Subtract	$HI \mid LO -= (\text{int})Rs * (\text{int})Rt$
MSUBU	Multiply-Subtract Unsigned	$HI \mid LO -= (\text{uns})Rs * (\text{uns})Rt$
MTC0	Move to Coprocessor 0	$CPR[0, n, Sel] = Rt$
MTHI	Move to HI	$HI = Rs$
MTLO	Move to LO	$LO = Rs$
MUL	Multiply with register write	$HI \mid LO = \text{Unpredictable}$ $Rd = ((\text{int})Rs * (\text{int})Rt)_{31..0}$
MULT	Integer Multiply	$HI \mid LO = (\text{int})Rs * (\text{int})Rd$
MULTU	Unsigned Multiply	$HI \mid LO = (\text{uns})Rs * (\text{uns})Rd$
NOP	No Operation (Assembler idiom for: SLL r0, r0, r0)	
NOR	Logical NOR	$Rd = \sim(Rs \mid Rt)$
OR	Logical OR	$Rd = Rs \mid Rt$
ORI	Logical OR Immediate	$Rt = Rs \mid \text{Immed}$
RDHWR	Read Hardware Register (if enabled by HWRE _{na} Register)	$Re = HWR[Rd]$

Note 1: This instruction is deprecated and should not be used.

28.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
 - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB C Compiler for Various Device Families
 - HI-TECH C for Various Device Families
 - MPASM™ Assembler
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
 - MPLAB ICD 3
 - PICKit™ 3 Debug Express
- Device Programmers
 - PICKit™ 2 Programmer
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

28.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - In-Circuit Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
 - Source files (C or assembly)
 - Mixed C and assembly
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

28.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

28.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, pre-processor, and one-step driver, and can run on multiple platforms.

28.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

28.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

28.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

TABLE 29-18: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 2.3V TO 3.6V)

AC CHARACTERISTICS		Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤T _A ≤+85°C for Industrial -40°C ≤T _A ≤+105°C for V-Temp					
Param. No.	Symbol	Characteristics ⁽¹⁾	Min.	Typical	Max.	Units	Conditions
OS50	FPLLI	PLL Voltage Controlled Oscillator (VCO) Input Frequency Range	4	—	5	MHz	ECPLL, HSPLL, XTPLL, FRCPLL modes
OS51	FSYS	On-Chip VCO System Frequency	60	—	120	MHz	—
OS52	TLOCK	PLL Start-up Time (Lock Time)	—	—	2	ms	—
OS53	DCLK	CLKO Stability ⁽²⁾ (Period Jitter or Cumulative)	-0.25	—	+0.25	%	Measured over 100 ms period

Note 1: These parameters are characterized, but not tested in manufacturing.

2: This jitter specification is based on clock-cycle by clock-cycle measurements. To get the effective jitter for individual time-bases on communication clocks, use the following formula:

$$EffectiveJitter = \frac{D_{CLK}}{\sqrt{\frac{SYSCLK}{CommunicationClock}}}$$

For example, if SYSCLK = 80 MHz and SPI bit rate = 20 MHz, the effective jitter is as follows:

$$EffectiveJitter = \frac{D_{CLK}}{\sqrt{\frac{80}{20}}} = \frac{D_{CLK}}{2}$$

TABLE 29-19: INTERNAL FRC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤T _A ≤+85°C for Industrial -40°C ≤T _A ≤+105°C for V-Temp				
Param. No.	Characteristics	Min.	Typical	Max.	Units	Conditions
Internal FRC Accuracy @ 8.00 MHz ⁽¹⁾						
F20	FRC	-2	—	+2	%	—

Note 1: Frequency calibrated at 25°C and 3.3V. TUN bits can be used to compensate for temperature drift.

TABLE 29-20: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤T _A ≤+85°C for Industrial -40°C ≤T _A ≤+105°C for V-Temp				
Param. No.	Characteristics	Min.	Typical	Max.	Units	Conditions
LPRC @ 31.25 kHz ⁽¹⁾						
F21	LPRC	-15	—	+15	%	—

Note 1: Change of LPRC frequency as V_{DD} changes.

FIGURE 29-23: EJTAG TIMING CHARACTERISTICS

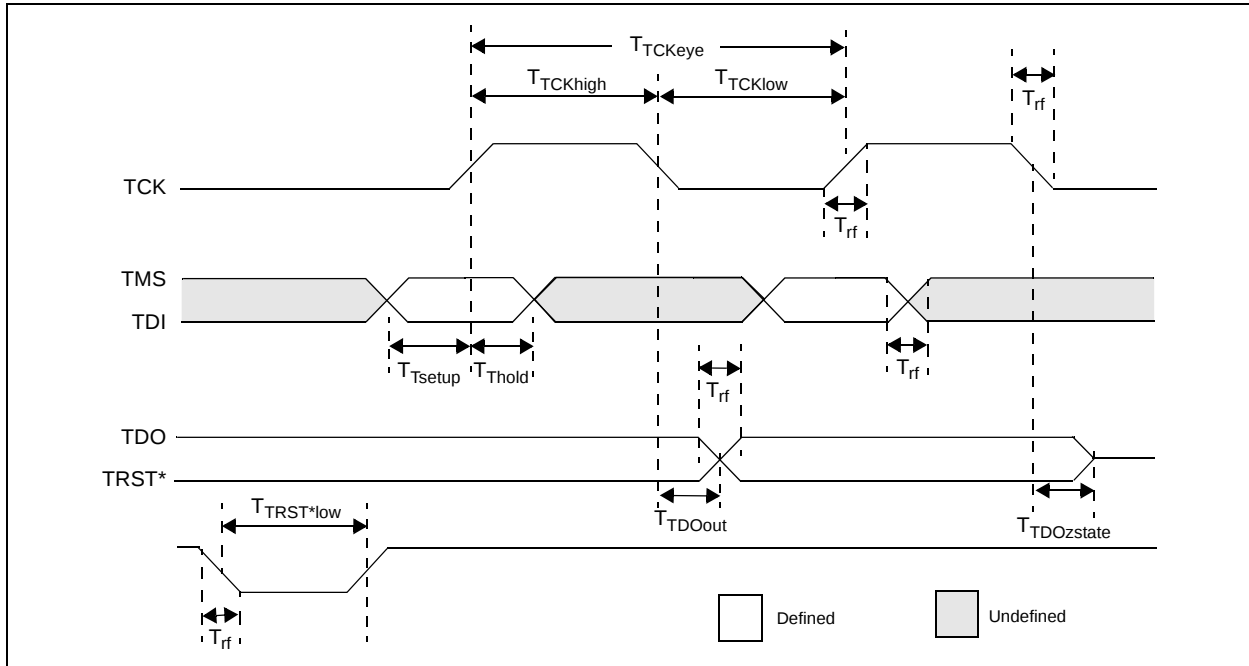


TABLE 29-41: EJTAG TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +105°C for V-Temp		
Param. No.	Symbol	Description ⁽¹⁾	Min.	Max.	Units	Conditions
EJ1	TTCKCYC	TCK Cycle Time	25	—	ns	—
EJ2	TTCKHIGH	TCK High Time	10	—	ns	—
EJ3	TTCKLOW	TCK Low Time	10	—	ns	—
EJ4	TTSETUP	TAP Signals Setup Time Before Rising TCK	5	—	ns	—
EJ5	TTHOLD	TAP Signals Hold Time After Rising TCK	3	—	ns	—
EJ6	TTDOOUT	TDO Output Delay Time from Falling TCK	—	5	ns	—
EJ7	TTDOZSTATE	TDO 3-State Delay Time from Falling TCK	—	5	ns	—
EJ8	TTRSTLOW	TRST Low Time	25	—	ns	—
EJ9	TRF	TAP Signals Rise/Fall Time, All Input and Output	—	—	ns	—

Note 1: These parameters are characterized, but not tested in manufacturing.

Revision G (April 2010)

The revision includes the following global update:

- Added Note 2 to the shaded table that appears at the beginning of each chapter. This new note provides information regarding the availability of registers and their associated bits.

This revision also includes minor typographical and formatting changes throughout the data sheet text. Major updates are referenced by their respective section in the following table.

TABLE A-2: MAJOR SECTION UPDATES

Section Name	Update Description
“High-Performance, General Purpose and USB 32-bit Flash Microcontrollers”	Updated the crystal oscillator range to 3 MHz to 25 MHz (see Peripheral Features:) Added the 121-pin Ball Grid Array (XBGA) pin diagram. Updated Table 1: “PIC32MX General Purpose – Features” and Table 2: “PIC32MX USB – Features” Added the following tables: - Table 3: “Pin Names: PIC32MX320F128L, PIC32MX340F128L, and PIC32MX360F128L, and PIC32MX360F512L Devices”, - Table 4: “Pin Names: PIC32MX440F128L, PIC32MX460F256L and PIC32MX460F512L Devices” Updated the following pins as 5V tolerant: - 64-pin QFN (USB): Pin 34 (V_{BUS}), Pin 36 (D-/RG3) and Pin 37 (D+/RG2) - 64-pin TQFP (USB): Pin 34 (V_{bus}), Pin 36 (D-/RG3), Pin 37 (D+/RG2) and Pin 42 (IC1/RTCC/INT1/RD8) - 100-pin TQFP (USB): Pin 54 (V_{bus}), Pin 56 (D-/RG3) and Pin 57 (D+/RG2)
Section 1.0 “Device Overview”	Updated the Pinout I/O Descriptions table to include the device pin numbers (see Table 1-1)
Section 2.0 “Guidelines for Getting Started with 32-bit Microcontrollers”	Updated the Ohm value for the low-ESR capacitor from less than 5 to less than 1 (see Section 2.3.1 “Internal Regulator Mode”). Labeled the capacitor on the V_{CAP}/V_{DDCORE} pin as CEFC in Figure 2-1. Changed 10 µF capacitor to CEFC capacitor in Section 2.3 “Capacitor on Internal Voltage Regulator (V_{CAP}/V_{CORE})”.
Section 4.0 “Memory Organization”	Updated all register map tables to include the “All Resets” column. Separated the PORT register maps into individual tables (see Table 4-21 through Table 4-34). In addition, formatting changes were made to improve readability.
Section 12.0 “I/O Ports”	Updated the second paragraph of Section 12.1.2 “Digital Inputs” and removed Table 12-1.
Section 22.0 “10-bit Analog-to-Digital Converter (ADC)”	Updated the ADC Conversion Clock Period Block Diagram (see Figure 22-2).
Section 26.0 “Special Features”	Extensive updates were made to Section 26.2 “Watchdog Timer (WDT)” and Section 26.3 “On-Chip Voltage Regulator” .

INDEX

A

AC Characteristics	161
Internal RC Accuracy	163
AC Electrical Specifications	
Parallel Master Port Read Requirements	186
Parallel Master Port Write Requirements.....	187
Parallel Slave Port Requirements	185
Assembler	
MPASM Assembler	148

B

Block Diagrams	
ADC Module	123
Comparator I/O Operating Modes.....	125
Comparator Voltage Reference	127
Connections for On-Chip Voltage Regulator.....	138
Input Capture	107
JTAG Compliant Application Showing	
Daisy-Chaining of Components	139
Output Compare Module.....	109
Reset System.....	87
RTCC	121
Type B Timer	37, 95, 105
UART	115
WDT.....	137
Brown-out Reset (BOR)	
and On-Chip Voltage Regulator	138

C

C Compilers	
MPLAB C18	148
Comparator	
Operation	126
Comparator Voltage Reference	
Configuring.....	128
CPU Module.....	31, 37
Customer Change Notification Service	209
Customer Notification Service.....	209
Customer Support.....	209

D

DC Characteristics	152
I/O Pin Input Specifications.....	157
I/O Pin Output Specifications.....	158
Idle Current (I _{IDLE})	154
Operating Current (I _{DD}).....	153
Power-Down Current (I _{PD})	155
Program Memory	159
Temperature and Voltage Specifications	152
Development Support	147

E

Electrical Characteristics.....	151
AC	161
Errata	19

F

Flash Program Memory	85
RTSP Operation.....	85

I

I/O Ports	101, 115
Parallel I/O (PIO).....	102
Internet Address.....	209

M

Microchip Internet Web Site.....	209
MPLAB ASM30 Assembler, Linker, Librarian	148
MPLAB Integrated Development Environment Software..	147
MPLAB PM3 Device Programmer	150
MPLAB REAL ICE In-Circuit Emulator System	149
MPLINK Object Linker/MPLIB Object Librarian	148

P

Packaging	191
Details.....	192
Marking	191
PIC32 Family USB Interface Diagram	100
Pinout I/O Descriptions (table).....	22
Power-on Reset (POR)	
and On-Chip Voltage Regulator	138

R

Reader Response.....	210
----------------------	-----

S

Serial Peripheral Interface (SPI) ...	87, 97, 111, 119, 121, 130
Software Simulator (MPLAB SIM)	149
Special Features.....	131

T

Timer1 Module.....	89, 95, 103, 105
Timing Diagrams	
10-bit Analog-to-Digital Conversion (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 0, SSRC<2:0> = 000) ..	183
10-bit Analog-to-Digital Conversion (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 1, SSRC<2:0> = 111, SAMC<4:0> = 00001).....	184
I2Cx Bus Data (Master Mode)	175
I2Cx Bus Data (Slave Mode)	177
I2Cx Bus Start/Stop Bits (Master Mode).....	175
I2Cx Bus Start/Stop Bits (Slave Mode).....	177
Input Capture (CAPx)	169
OC/PWM	170
Output Compare (OCx)	169
Parallel Master Port Write.....	186, 187
Parallel Slave Port.....	185
SPIx Master Mode (CKE = 0)	171
SPIx Master Mode (CKE = 1)	172
SPIx Slave Mode (CKE = 0)	173
SPIx Slave Mode (CKE = 1)	174
Timer1, 2, 3, 4, 5 External Clock	167
Transmission (8-bit or 9-bit Data)	116
UART Reception with Receive Overrun	117
Timing Requirements	
CLKO and I/O	164
Timing Specifications	
I2Cx Bus Data Requirements (Master Mode).....	175
I2Cx Bus Data Requirements (Slave Mode).....	178
Output Compare Requirements.....	169
Simple OC/PWM Mode Requirements	170
SPIx Master Mode (CKE = 0) Requirements.....	171
SPIx Master Mode (CKE = 1) Requirements.....	172
SPIx Slave Mode (CKE = 1) Requirements.....	174

V

VCORE/VCAP Pin	138
Voltage Reference Specifications.....	160
Voltage Regulator (On-Chip)	138