Section 1. Introduction

HIGHLIGHTS

This section of the manual contains the following topics:

1.1 INTRODUCTION

Microchip's PIC32MX series of 32-bit microcontrollers is designed to fulfill customers' requirements for enhanced features and performance for their MCU-based applications.

Common attributes among all devices in the PIC32MX series are:

- Pin, peripheral and source code compatibility with the PIC24F128GAXXX family
- MIPS32® M4K™ processor core
- Common development tools

1.2 OBJECTIVE OF THIS MANUAL

This manual describes the PIC32MX series of 32-bit microcontrollers. It explains the family architecture and operation of the peripheral modules, but does not cover the specifics of each device in the family. Users should refer to the respective device's data sheet for device-specific details, such as:

- Pinout and packaging details
- Memory map
- List of peripherals included on the device, including multiple instances of peripherals
- Device-specific electrical specifications and characteristics

1.3 DEVICE STRUCTURE

The PIC32MX architecture has been broken down into the following functional blocks:

- MCU Core
- System Memory
- System Integration
- Peripherals

1.3.1 MCU Core

The MCU core consists of these essential basic features.

- 32-bit RISC MIPS32 M4K Core
- Single Cycle ALU
- Load-Store Execution Unit
- 5-Stage Pipeline
- 32-bit Address and 32-bit Data Buses
- Two 32-element, 32-bit General Purpose Register Files
- FMT – Fixed Mapping Translation Memory Management
- FMDU – Fast-Multiply-Divide Unit
- MIPS32® Compatible Instruction Set
- MIPS16e™ Code Compression Instruction Set Architecture Support

The CPU section of this manual discusses the PIC32MX MCU core.

1.3.2 System Memory

The system memory provides on-chip nonvolatile Flash memory and volatile SRAM memory, featuring user and protected kernel-segment-partitioning for real-time operating systems. The following sections of this manual discuss the PIC32MX system memory:

- Section 3. Memory Organization
- Section 5. Flash Programming

Flash Memory Technology

- The Flash can be used for program memory or data.
- The Flash allows program memory to be electrically erased or programmed under software control during normal device operation.
- The PIC32MX series has full-speed execution directly from program Flash through the use of on-chip prefetch buffering by the Prefetch module.
- The Flash has the capability to page erase, word or row program.

1.3.3 System Integration

System integration consists of a comprehensive set of modules and features that tie the MCU core and peripheral modules into a single operational unit. System integration features also provide these advantages:

- Decreased system cost, by bringing traditionally off-chip functions into the microcontroller
- Increased design flexibility, by adding a wider range of operating modes
- Increased system reliability, by enhancing the ability to recover from unexpected events

The following sections of this manual discuss the PIC32MX system integration:

- Section 3. Memory Organization
- Section 4. Prefetch Module
- Section 5. Flash Programming
- Section 6. Oscillator
- Section 7. Resets
- Section 8. Interrupts
- Section 9. Watchdog Timer and Power-up Timer
- Section 10. Power-Saving Modes
- Section 31. Direct Memory Access (DMA) Controller with programmable Cyclic Redundancy Check (CRC)
- Section 32. High-Level Integration (Configuration, Code Protection and Voltage Regulation)
- Section 33. Device Programming, Debugging, In-Circuit and In-Circuit Testing

1.3.4 Peripherals

The PIC32MX devices have many peripherals that allow it to interface with the external world. The following sections of this manual discuss the PIC32MX peripherals:

- Section 12. I/O Ports
- Section 13. Parallel Master Port
- Section 14. Timers
- Section 15. Input Capture Module
- Section 16. Output Compare/Pulse Width Modulation (PWM) Module
- Section 17. 10-bit A/D Converter
- Section 19. Comparator Module
- Section 20. Comparator Voltage Reference Module
- Section 21. UART Module
- Section 23. SPI Module
- Section 24. I^2C™ Module
- Section 27. USB OTG
- Section 29. Real-Time Clock/Calendar (RTCC) Module

1.4 DEVELOPMENT SUPPORT

Microchip offers a wide range of development tools that allow users to efficiently develop and debug application code. Microchip's development tools can be broken down into four categories:

- Code generation
- Hardware/software debug
- Device programmer
- Product evaluation boards

A full description of each of Microchip's development tools is given in **Section 29.0 "Development Support"**. As new tools are developed, the latest product briefs and user guides can be obtained from the Microchip web site (www.microchip.com) or from local Microchip Sales Offices.

Microchip offers other references and support to speed the development cycle. These include:

- Application notes
- Reference designs
- Microchip web site
- Local sales offices with field application support
- Corporate Applications support line
- Getting Stated guide
- "How to" brochures
- Masters Conferences
- Webinars
- Design Centers

These can all be found on the Microchip web site. Also, the Microchip web site lists other sites that may provide useful references.

1.5 STYLE AND SYMBOL CONVENTIONS

Throughout this document, certain style, format, and font conventions are used to signal particular distinctions for the affected text. Table 1-1 lists these conventions, the MCU-industry-specific symbols, and non-conventional word definitions and abbreviations used in this manual.

Located at the rear of this document, a glossary provides additional word and abbreviation definitions for content used in this manual.

1.5.1 Document Conventions

Table 1-1 defines some of the symbols, terms and typographic conventions used in this manual.

Table 1-1: Document Conventions

SYMBOL AND TERM CONVENTIONS:	
Convention	**Description**
set	To force a bit/register to a value of logic '1'.
clear	To force a bit/register to a value of logic '0'.
reset	1. To force a register/bit to its default state. 2. A condition in which the device places itself after a device Reset occurs. Some bits will be forced to '0' (such as interrupt enable bits), while others will be forced to '1' (such as the I/O data direction bits).
: (colon)	Specifies a range or concatenation of registers/bits/pins. Concatenation order (left to right) usually specifies a positional relationship (MSb to LSb, higher to lower). For example, TMR3:TMR2 indicates the concatenation of two 16-bit registers to form a 32-bit timer value, with the value of TMR3 representing the most significant half-word of the value.
< >	Specifies a bit location or range of locations within a particular register or field of similarly-named bits. For example, PTCON<2:0> specifies the range of the 3 Least Significant bits of the register PTCON.
MSb LSb	Most Significant bit and Least Significant bit.
MSB, LSB	Most Significant Byte, Least Significant Byte. (A Byte is 8-bits wide.)
mshw, lshw	Most Significant half-word and least significant half-word A Half-Word is 16-bits wide
msw, lsw	Most Significant Word and Least Significant Word. (A Word is 32-bits wide.)
0xnn	Designates the number 'nn' in the hexadecimal number system. This convention is used in code examples, and is equivalent to the notation 'nnh' used in text. For example, 0x13 is equivalent to 13h.
FONT CONVENTIONS:	
Arial Font	The standard font used for all text, figures and tables within this manual. Other fonts, as described below, are used to set off mathematical and logical expressions, or device instruction code, from descriptive text.
Courier New Font	Within text, this font is used for contrast with the standard text font and specifically denote the following: 1. an instruction set mnemonic or assembler code fragment. 2. the binary value of a bit, range of bits, or a register. 3. the logical state of a digital signal. Within code examples, this font is used exclusively to denote an assembly or high-level language instruction sequence.
Times New Roman Font	The standard font for mathematical expressions and variables.
GRAPHIC CONVENTIONS:	
Note	A note presents information that requires emphasis: either to help users avoid a common pitfall, or to make them aware of operating differences between some device family members. A note is usually in a shaded box, unless it is used in a bit description, or as a table or diagram footnote. **Note:** This is a Note in a shaded note box.
Register Cells	A bit name that appears in a grayed-out cell of a register signals that the bit is not relevant to the peripheral module described in that particular section of the manual. FRZ

1.5.2 Electrical Specifications

Throughout this manual, there are references to electrical specifications and their parameter numbers. Table 1-2 shows the parameter numbering convention for PIC32MX devices. A parameter number represents a unique set of characteristics and conditions that is consistent between every data sheet, though the actual parameter value may vary from device to device.

This manual describes a family of devices and, therefore, does not specify the parameter values. To determine the parameter values for a specific device, users should refer to the “Electrical Specifications” section of that device’s data sheet.

Table 1-2: Electrical Specification Parameter Numbering Convention

Parameter Number Format	Comment
DXXX	DC Specification
AXXX	DC Specification for Analog Peripherals
XXX	Timing (AC) Specification
PDXXX	Device Programming DC Specification
PXXX	Device Programming Timing (AC) Specification

Legend: XXX represents a parameter number.

1.6 RELATED DOCUMENTS

Microchip, as well as other sources, offers additional documentation to aid you as you develop PIC32MX-based applications. The list below contains the most common documentation, but other documents may also be available. Please check the Microchip web site (www.microchip.com) for the latest published technical documentation.

1.6.1 Microchip Documentation

The following PIC32MX documentation is available from Microchip. Many of these documents provide application-specific information that gives actual examples of using, programming, and designing with PIC32MX microcontrollers.

1. *PIC32MX Family Reference Manual*
   The family reference manual describes the PIC32MX architecture and operation of the peripheral modules, but does not cover the specifics of each device in the family.

2. PIC32MX Data Sheets
   The data sheets contain device-specific information, such as pinout and packaging details, electrical specifications and memory maps.

3. *PIC32MX Programming Specification*
   The programming specifications contain detailed descriptions of, and electrical and timing specifications for, the programming process. Both In-Circuit Serial Programming™ (ICSP™) and Enhanced ICSP are described in detail.

1.6.2 Third-Party Documentation

Microchip does not review third-party documentation for technical accuracy, but these references may be helpful to understand operation of the devices. The Microchip web site may have information on these third-party documents.

1.7 REVISION HISTORY

Revision A (September 2007)

This is the initial version of this document.

Revision B (October 2007)

Updated document to remove Confidential status.

Revision C (April 2008)

Revised status to Preliminary; Revised Section 1.1.

NOTES:

