PIC18F to PIC24F Migration: An Overview

INTRODUCTION

The PIC24F architecture specification was created as a powerful extension of Microchip's existing RISC microcontroller portfolio. By giving users an even greater range of options for computational power and rich peripheral sets, it allows users to grow their applications. At the same time, the architecture was tailored to keep as much of the existing PICmicro® MCU feature set and nomenclature as possible, making it easy for applications to make the jump.

This migration document highlights the similarities and differences between the PIC18F and PIC24F device families, and shows the general principles for migrating PIC18F applications to PIC24F devices. Throughout this document, it is assumed that the application to be ported is based on a member of one of the later PIC18F device families, such as the popular PIC18F8722 family, with a complete feature set and the latest version of nanoWatt Technology of all PIC18F devices. The target device is assumed to be in the PIC24FJ128GA general purpose family, which is the first generation of PIC24F general purpose devices. However, the general guidelines represented here can be applied to migrating any PIC18F device-based application to a PIC24F platform.

To present such a comprehensive overview, it was decided to divide the material into two major sections, each with a slightly different approach. The first section compares and contrasts the core architectural differences between the families, highlighting the major differences. Because the changes can greatly change the overall structure of an application, the focus is not on how to do specific tasks, but what larger changes need to be considered in migration. The specific modifications required are left to the user's professional judgment.

The second section discusses the peripherals that PIC18F and PIC24F architectures have in common. Like the core, module features are compared. Here, it is possible to also compare the steps needed to make each module run and give task-specific information. Only those peripheral features that are available in both architectures are presented here.

Users are encouraged to review the PIC24F device data sheets for information on the new modules, and how they may be used in applications that are candidates for migration.

Note 1: This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

2: The user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the oscillator mode may be required.

Contents

PIC24F CORE ARCHITECTURE

CPU Core

Aside from its 16-bit data size, the PIC24F architecture is significantly different from the PIC18F architecture. The instruction word size, instruction clocking scheme, stack implementation and core registers are very different from PIC18F implementations. Other hardware features have been added to enhance processing performance. Changes are summarized in Table 1.

TABLE 1: COMPARISON OF PIC18F AND PIC24F CPU CORE FEATURES

Feature	PIC18F	PIC24F
Instruction Size	16 bits	24 bits
Instruction Clocking	TCY = FOSC/4	TCY = FOSC/2
Working Registers	1 (W, WREG)	16 (W0-W15)
Status Registers	One (STATUS)	Two (STATUS and CORCON)
Stack	Hardware, 31 levels	Software
Hardware Multiplier	8 x 8	17 x 17
Hardware Divider	No	Hardware assisted division using `DIV` and `REPEAT`
Bit Shifting/Rotation	Single bit, left or right, rotation only	Barrel shifting up to 15 bits, left or right, shift or rotate
Program Space Visibility (PSV)	No	Yes

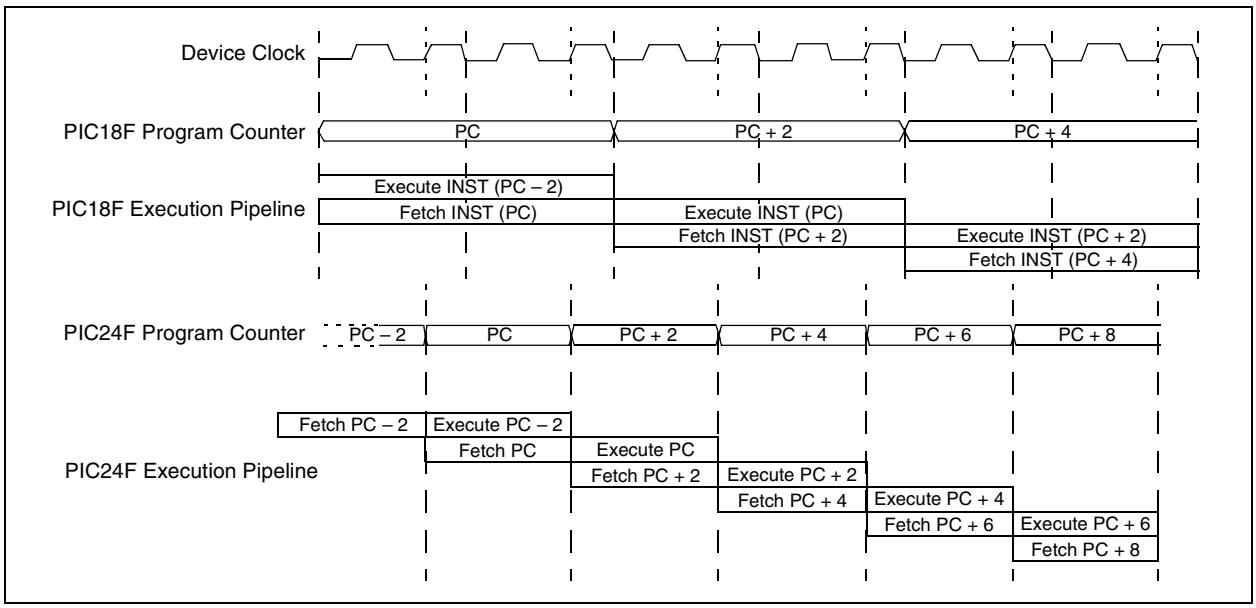
INSTRUCTION SIZE

All PIC18F devices use a 16-bit (single-word) instruction. PIC24F devices use a 24-bit instruction. A more detailed discussion on instructions is presented in the **"Instruction Set"** section.

INSTRUCTION CLOCKING

PICmicro microcontrollers execute instructions at a rate that is a simple fraction of the clock speed, designated as TCY. For the most part, PIC18F and PIC24F devices perform most instructions with a fetch step and an execution step of 1 TCY each. For PIC18F devices, each TCY interval represents 4 clock cycles, with the PC being incremented on falling edges. In contrast, the TCY for PIC24F devices is 2 clock cycles, with the PC being incremented on leading clock edges. The differences in instruction pipelining are shown in Figure 1. This difference in instruction execution is one of the biggest differences between the two architectures, and needs to be taken into consideration for any peripherals that use the instruction clock speed as a base for their timing.

FIGURE 1: COMPARISON OF PIC18F AND PIC24F INSTRUCTION EXECUTION PIPELINES



W REGISTERS

All PIC18F devices have a single working register, known as W (sometimes, WREG). W is used as an operand in almost all instructions, and is either the source or destination for all instructions.

The PIC24F architecture increases the number of working registers to 16, named W0 through W15. Most instructions allow for the selection of any one of these as source and/or destination for the operation, although several instructions only allow the use of WREG (W0).

Several of the registers have special functions. W0 and W1 are the working registers for the hardware-assisted divider, and are required during divide (`DIV`) instructions. W2 and W3 are used by the `MUL` instruction as result registers. W14 and W15 are used for a Frame Pointer and Stack Pointer, respectively.

STATUS REGISTERS

PIC18F devices have one ALU status register, named STATUS. It contains the normal flag bits for binary digital math operations: Carry, Digit Carry, Zero state, Negative state and Overflow. All flags are writable as well as readable.

PIC24F devices use two status registers, which provide these ALU flags as well as CPU control bits. The STATUS register contains the standard ALU flags already listed; it also contains the RA flag to indicate an active Repeat loop and the IPL2:IPL0 bits to set the CPU interrupt priority level.

In addition to STATUS, PIC24F devices also have the CORCON register. It contains the IPL3 bit, which effectively enables or disables peripheral interrupts and the PSV bit, which enables the Program Space Visibility feature (see **"Data Memory Space"** on page 11 for more information).

STACK

PIC18F devices implement the program counter stack in hardware. The stack has a total of 32 levels, with only the top layer being accessible to software. Stack pushes and pops are executed as instructions. An optional, non-maskable hardware trap (Reset) can be enabled for stack overflow and underflow events.

In contrast, PIC24F devices implement a software stack in data RAM. The stack starts at RAM address 0800h and grows towards higher addresses. W15 serves as the Stack Pointer and W14 is used as a Frame Pointer for higher level language applications. The upper limit of the stack is defined in software using the SPLIM register. Stack Pointer underflow and overflow traps are generated when the Stack Pointer is decremented below 0800h, or above the limit set by SPLIM.

MULTIPLIER

Both PIC18F and PIC24F devices include a dedicated hardware multiplier in their ALUs. The PIC18F multiplier is an 8 x 8 unit, capable of supporting signed, unsigned and mixed-sign operations. The PIC24F multiplier is a 17 x 17 unit and can also support signed, unsigned and mixed-signed operations. PIC18F multipliers can perform 8-bit unsigned operations in a single instruction cycle, while PIC24F multipliers can perform a 16-bit by 16-bit signed or mixed operation in a single cycle.

DIVIDER

PIC18F devices do not provide any hardware support for division. Typical divide operations (signed 16/16 or 16/8) performed, using the standard PIC18F math library, can take up to 38 instruction cycles to execute.

PIC24F devices do not have a hardware divider per se. Instead, the PIC24F ALU is configured in hardware to support a divide instruction, `DIV`. Together with the `REPEAT` control instruction, `DIV` allows the ALU to automatically execute the iterative division process as a simple sequence instead of a long algorithm. `DIV` supports several forms of 32/16 and 16/16 divides, including fixed-point and fractional, and performs executions in 19 instruction cycles.

BIT SHIFTS AND ROTATES

The PIC18F ALU is capable of performing single bit position rotations, with or without carry, in either direction. Any byte operation that requires rotations of more than one position will require an equal number of instruction cycles to perform the operation. Word length operations are potentially longer and more complex, as bits need to be carried from one byte of data to another.

The PIC24F ALU is equipped with a barrel shifter that allows data to be either rotated or shifted (without wrap-around), with or without carry, in either direction. This permits shifts of multiple bit positions in a single instruction cycle. Data can be rotated either as bytes for up to 7 bits at a time, or as words for up to 15 bits at a time.

Instruction Set

The PIC24F instruction set architecture represents a considerable extension of its PIC18F predecessor. While barely larger in terms of base instructions, it makes substitutions of selected instructions to enhance the performance, while expanding the scope of addressing modes and data type operations. Through all of this, the PIC24F instruction set maintains the highly orthogonal structure of previous PICmicro architectures.

The main differences between the PIC18F and PIC24F instruction sets are presented in Table 2.

TABLE 2: COMPARISON OF MAJOR INSTRUCTION SET FEATURES

Feature	PIC18F	PIC24F
Instruction Size	16 bits	24 bits
Base Instructions	75	76
Supported Data Types	Byte	Byte, Word, Double Word
Operand Support	Up to binary (b = a + b)	Up to trinary (a + b = c)
Addressing Modes	Direct, Indirect (5)	Direct, Indirect (6)
Indirect Addressing Type	Uses FSR Pointer register sets; limit of 3 pointers	Uses any of the W registers as pointers with instruction-based manipulation; limit of 16
Data Space Addressing	Short literal, limited to a single bank at one time; full literal on few instructions	Short literal for entire Near Data Space; indirect address for entire space

CHANGES FROM PIC18F INSTRUCTION SET

The PIC24F instruction set implements several features that enhance performance for math operations, and extend the ability to handle high-level languages with more complex stack and pointer requirements. These are summarized in Table 3.

In addition, several PIC18F instructions do not have an exact single instruction equivalent in the PIC24F architecture. Most of these instructions have either single or pairs of PIC24F instructions that have similar effect, but because of differences in the STATUS register, results are not exactly equivalent. They are:

- CPFSEQ
- CPFSGT
- CPFSLT
- DCFSNZ
- DECFSZ
- INCFSZ
- INFSNZ
- MOVSF
- MOVSS
- PUSHL
- SUBULNK
- TSTFSZ
- MOVFF (all modes except indirect to indirect)
- SWAPF (all modes except with WREG)

Even with these changes, most PIC18F assembler instructions have a single-cycle equivalent in the PIC24F assembler. A comprehensive list is provided in **Appendix A: "Mapping PIC18F to PIC24F Instructions"**.

TABLE 3: NEW OR SIGNIFICANTLY MODIFIED PIC24F INSTRUCTIONS

PIC24F Instruction	Description
DIV	Divide two numbers (signed or unsigned, 16/16 or 32/16)
LNK and UNLK	Link or unlink Frame Pointer (W14)
LSR and ASR	Logical or arithmetic shift right by 1 to 16 bits, by either a literal or variable value
MUL.SS, MUL.SU, MUL.US	Specific instructions for multiplying signed and unsigned numbers
PUSH and POP	Both instructions now include source or destination arguments, allowing the stack to store values other then the current PC
REPEAT	Repeat the next instruction a specified number of times
SL	Shift left by 1 to 16 bits by either a literal or variable value

SUPPORTED DATA TYPES

Aside from bit-oriented instructions, PIC18F instructions operate exclusively on single byte data. Any data that is longer than a single byte is handled by concatenating bytes in the proper order.

As 16-bit devices, PIC24F instructions are designed to handle data in terms of 16-bit words. In addition, most instructions are capable of handling single byte and double word (32-bit) data objects. Invoking this feature is generally done by using the suffix ".b" (for bytes) or ".d" (for double words) after the instruction mnemonic. Data alignment within the memory space is automatically adjusted and maintained according to the data type selected.

OPERAND SUPPORT

The PIC18F instruction set supports both unary and binary operations. That is to say, an instruction can function with a single argument operating on the contents of one register (e.g., increment the WREG), or with two arguments storing the results in one of the two registers (e.g., adding the contents of a register to WREG and storing the sum in WREG).

The PIC24F instruction set also supports trinary operations; that is, an operation can have three arguments, specifying not only two operands but also an independent address for the result. Trinary operations are supported for most arithmetic and logic instructions.

ADDRESSING MODES

The biggest difference between the PIC18F and PIC24F instruction sets is how the instructions specify addresses in the data memory space. While the methods are equivalent, the PIC24F version allows more options for more of its instructions.

Direct Addressing

While PIC18F instructions can directly address any register in the data space, very few specify a full address as part of the instruction. Instead, most use an 8-bit literal value to specify an address within one of 16 banks of 256 bytes within the space. The bank in question is separately selected by the Bank Select Register (BSR).

PIC24F instructions are designed to be able to literally address any register within the lowest 8 Kbytes of the data space (also known as the Near Data Space). This allows immediate access to all SFRs. Implemented memory areas above the top of the Near Data Space are accessible by indirect addressing, as described below.

Indirect Addressing

PIC18F instructions perform indirect addressing through pointers and virtual registers. Three 12-bit pointers (FSR0 through FSR2) are loaded with appropriate values; then, one of five virtual registers associated with that pointer is used as an argument for an instruction, either as an address or an offset for an address. Depending on which of the registers is used, the value of the pointer can also be automatically incremented or decremented. Only these three pointers and their associated registers can be used for indirect addressing.

In contrast, PIC24F architecture increases the number of options for the addressing mode. Any one of the W registers can be used as a pointer for indirect addressing. Instead of using a virtual register name to determine the pointer and its operation, the MPLAB® IDE assembler language uses specific syntax conventions to indicate when a W register is being used as a pointer or offset, and what increment or decrement operation is to be done to the register. The PIC24F Instruction Set Architecture (ISA) also offers a pre-decrement option not available with PIC18F. An additional feature of the PIC24F increment/decrement options is how they work with the data type specified by the instruction. If the instruction is executed as a byte type instruction, the register will be incremented or decremented by one. This aligns the pointer to the next byte in memory which is not always the next memory address. Word-oriented instructions will increment or decrement the register by two and maintain proper word alignment.

The differences between architectures in indirect addressing modes and syntax are summarized in Table 4. A comparison of how to use these addressing modes is shown in Example 1.

TABLE 4: INDIRECT ADDRESSING IN PIC18F AND PIC24F INSTRUCTION SETS

Indirect Addressing Mode	PIC18F Virtual Register Operand	PIC24F Equivalent Syntax	Wn (or FSR)	
			After Byte Instruction	After Word Instruction
No Modification	INDFx	[Wn]	Wn = Wn	Wn = Wn
Pre-Increment	PREINCx	[++Wn]	Wn = Wn + 1	Wn = Wn + 2
Pre-Decrement	N/A	[--Wn]	Wn = Wn – 1	Wn = Wn – 2
Post-Increment	POSTINCx	[Wn++]	Wn = Wn + 1	Wn = Wn + 2
Post-Decrement	POSTDECx	[Wn--]	Wn = Wn – 1	Wn = Wn – 2
Register Offset	PLUSWx	[Wn + Wb]	Wn = Wn	Wn = Wn

EXAMPLE 1: COMPARISON OF INDIRECT ADDRESSING TECHNIQUES

Typical PIC18F Code Sequence:

```
LoopSetup   MOVLW   0x10, LoopCount          ;set up loop counter
            LFSR    0, myArray0             ;set up pointer 0
            LFSR    1, myArray1             ;set up pointer 1
CopyLoop    MOVFF   POSTINC0, POSTINC1      ;copy myArray1 to myArray0
            DECFSZ  LoopCount, F            ;decrement loop counter
            BRA     CopyLoop                ;loop
```

Equivalent PIC24F Code Sequence:

```
LoopSetup   MOV     #myArray0, W0           ;set up pointer 0
            MOV     #myArray1, W1           ;set up pointer 1
CopyLoop    REPEAT  #15                     ;loop 16 times
            MOV     [W1++], [W0++]          ;copy array1 to array
```

RAW Dependencies

The increased flexibility in addressing modes creates a few situations where Read-After-Write (or RAW) dependencies may be created. RAW dependencies exist when a variable needed for an instruction that is being fetched stage has not been written back to the register yet. These occur only in select situations, generally when a register used as a destination for one instruction is also an argument in an immediately following instruction, and are listed in Table 5. The PIC24F CPU core includes “look ahead” detection for these RAW hazards, and introduces one or more stall (NOP) cycles between instructions to avoid execution errors.

TABLE 5: SUMMARY OF READ-AFTER-WRITE DEPENDENCY RULES

Destination Addressing Mode Using Wn	Source Addressing Mode Using Wn	Required Stall (CPU cycles added to instruction time)	Examples (Wn = W2)
Direct	Direct	None	`ADD.w W0, W1, W2` `MOV.w W2, W3`
Indirect	Direct	None	`ADD.w W0, W1, [W2]` `MOV.w W2, W3`
Indirect	Indirect	None	`ADD.w W0, W1, [W2]` `MOV.w [W2], W3`
Indirect	Indirect with pre/post-modification	None	`ADD.w W0, W1, [W2]` `MOV.w [W2++], W3`
Indirect with pre/post-modification	Direct	None	`ADD.w W0, W1, [W2++]` `MOV.w W2, W3`
Direct	Indirect	1	`ADD.w W0, W1, W2` `MOV.w [W2], W3`
Direct	Indirect with pre/post-modification	1	`ADD.w W0, W1, W2` `MOV.w [W2++], W3`

Indirect	Indirect	1	`ADD.w W0, W1, [W2]`
			`MOV.w [W2], W3`
			`; W2=04h (mapped W2)`

Indirect	Indirect with pre/post-modification	1	`ADD.w W0, W1, [W2]`
			`MOV.w [W2++], W3`
			`; W2=04h (mapped W2)`

Indirect with pre/post-modification	Indirect	1	`ADD.w W0, W1, [W2++]` `MOV.w [W2], W3`
Indirect with pre/post-modification	Indirect with pre/post-modification	1	`ADD.w W0, W1, [W2++]` `MOV.w [W2++], W3`

Memory Map and Program Memory

Both PIC18F and PIC24F architectures use the same general schema for their program memory spaces. Aside from the self-evident differences in width, PIC24F devices also incorporate a larger addressing range and enhanced visibility features in data space. The organization of the space and the location of non-program memory features also differ somewhat, and must be considered when porting an application.

The key differences between the memory organization of PIC18F and PIC24F devices are presented in Table 6.

TABLE 6: COMPARISON OF PIC18F AND PIC24F PROGRAM MEMORY ARCHITECTURES

Feature	PIC18F	PIC24F
Organization	16-bit, byte addressable	24-bit, word addressable
Total Addressable Range	4 Mbytes (22-bit magnitude)	16 Mbytes (24-bit magnitude)
Maximum Available User Program Space (upper boundary address)	2 Mbytes (FFFFFh)	8 Mbytes (7FFFFFh)
Boot Block Support	Most devices	No
Interrupt/Reset/Trap Vectors	00h, 08h, 18h	00h to 1FFh
Configuration Word Locations	300000h to 30000Fh	Last 2 implemented locations in program memory
Device ID Locations	3FFFFE and 3FFFFFh	FF0000h and FF0002h

ORGANIZATION

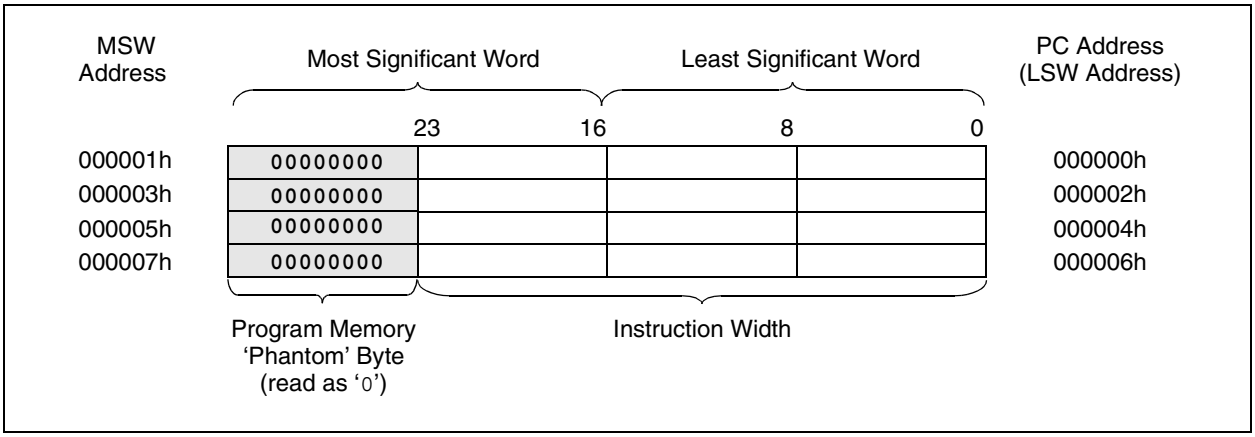
The PIC18F program space is organized as 16-bit words, but is addressable in terms of bytes. This means that the upper or lower byte of any word can be individually addressed by a pointer. To maintain word alignment for code execution, the memory space is aligned on the Least Significant Byte (LSB) of each word, and the program counter increments by 2 during normal execution.

The PIC24F program space has a different, but parallel structure. It is physically organized as 24 bits wide, but is addressed as 16-bit words. Thus, an instruction is considered to be two words, not 3 bytes. Each word can be individually addressed with even addresses representing the lower word of an instruction and odd addresses the upper word. To maintain instruction alignment, the memory space is aligned on even words; the program counter increments by 2 during normal execution.

Since a PIC24F instruction is three bytes wide, the Most Significant Byte of an instruction finds itself alone in the most significant word of the instruction as it is stored in memory. To maintain word alignment when reading from, or writing to program memory, a “phantom byte” of 00h is added before the MSB to give it the proper word length (Figure 2). The value reflects that this byte is not actually implemented in physical program memory.

Even with its word-aligned addressing, any single byte within the program space can be individually read or written. The PIC24F `TBLRD` and `TBLWT` commands are extended to allow an individual upper or lower byte of any word to be accessed. Keep in mind that the upper byte of any upper word (odd address) will always be 00h, and cannot be written to, for reasons previously discussed.

FIGURE 2: PIC24F PROGRAM MEMORY ORGANIZATION



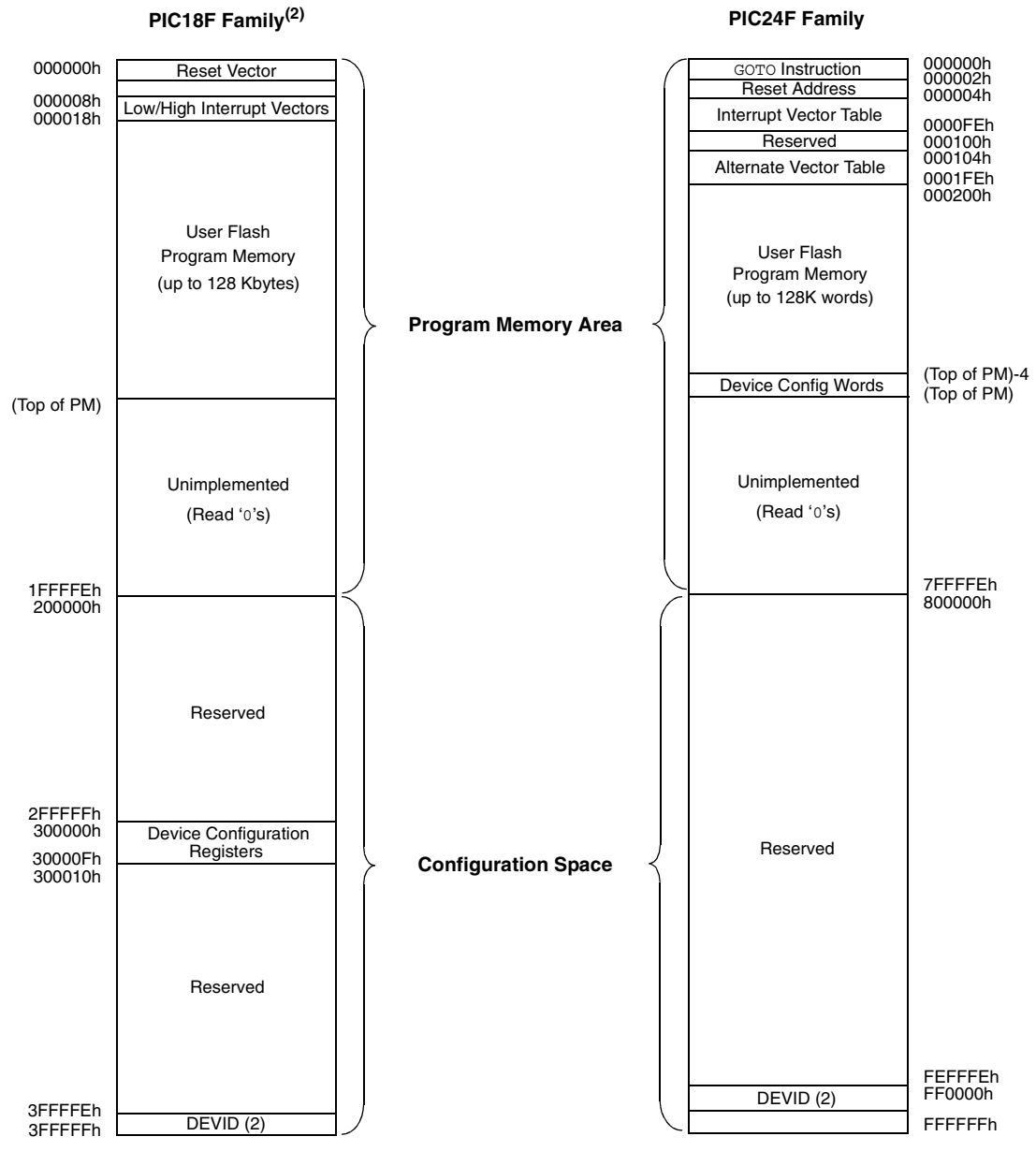
ADDRESSABLE AND USER-AVAILABLE RANGE

Both architectures base the size of their program space entirely on the size of the program counter. PIC18F devices use a 22-bit program counter for a total addressable space of 4 Mbytes. PIC24F devices use a 24-bit counter for a total range of 16 Mbytes.

In addition, both architectures reserve the upper half of the addressable program space as “configuration space”; this is largely unimplemented addresses, with several implemented areas for device configuration, identification and programming. Thus, the total maximum available space for program memory is 2 Mbytes for PIC18F devices, and 8 Mbytes for PIC24F devices. It is worth noting that no device in either family entirely implements the full available range of program memory space.

The program spaces for the two architectures are shown in comparison in Figure 3.

FIGURE 3: COMPARISON OF PIC18F AND PIC24F PROGRAM MEMORY SPACES(1)



Note 1: Memory areas are not shown to scale. Scales differ for PIC18F and PIC24F maps.

2: Only the elements of program space common to all PIC18F devices are shown. Later features, such as Boot Blocks and Flash Configuration Words (PIC18FXXJ Flash devices only) have been omitted for clarity.

INTERRUPT/RESET/TRAP VECTORS

In PIC18F devices, three addresses at the bottom of program memory are reserved for hardware vectors. Application code for these devices places destination addresses for hardware Resets, high-level Interrupt Service Routines (ISRs) and low-level ISRs in 00h, 08h and 18h, respectively.

In PIC24F devices, the reserved range is much larger: from 00h to 1FFh. Location 00h is still reserved for the hardware Reset vector. Addresses 06h through FFh are reserved for the main interrupt vector table, which contains 118 separate hardware address vectors (including 8 non-maskable hardware traps). Addresses 100h through 1FFh contain an alternate interrupt vector table, also of 118 hardware address vectors, that is reserved for use in Emulation mode.

CONFIGURATION WORDS AND DEVICE IDS

Both PIC18F and PIC24F devices locate their factory device IDs in the configuration space above the program memory area. PIC18F Configuration registers are also located in configuration space, whereas PIC24F Configuration registers are located in program memory. PIC18F devices locate their Configuration Words at 300000h through 30000Fh, and device IDs at 3FFFFEh and 3FFFFFh. PIC24F devices locate their Configuration Words at the last two addresses in implemented program memory, and device IDs at FF0000h and FF0002h.

BOOT BLOCK SUPPORT

Many PIC18F devices allow users to reserve and independently code-protect a small area at the base of the user program space (depending on the device, between 1 and 4K words) for an independent bootloader program.

As of this writing, a similar feature is not implemented in PIC24F devices. This absence does not rule out the use of bootloader programs; it only means that a dedicated and separately protectable block of program memory is not available.

EXTERNAL MEMORY MAPPING

The PIC18F architecture supports the use of external memory devices as program memory. This feature, known as the External Memory Interface (EMI), is actually implemented as part of the CPU core's address bus management system. The EMI allows the program address bus to address and retrieve data from off-chip devices as if they were part of the 2 Mbyte, on-chip program space. Multiple operating modes are implemented, such as using off-chip memory exclusively as program memory. Because of the I/O requirements, the EMI is only available on PIC18F devices with 80 or more pins.

As of this writing, PIC24F devices do not offer this feature. Program size is limited to that of on-chip memory.

FLASH MEMORY OPERATION

Many PIC18F and PIC24F devices use self-programmable Flash technologies to implement program memory. The general schema of memory operations and how they are performed are very similar, such as:

- the ability to self-program some or all of the Flash memory during run time
- the ability to program or reprogram a device already embedded in the application through a 5-wire serial interface (In-Circuit Serial Programming™ or ICSP™), or during device operation (Enhanced In-Circuit Serial Programming or EICSP)
- the use of erase and write blocks of defined sizes, and erase-before-write programming algorithms
- the self-timed write process after filling a write buffer block, which stalls CPU execution for one or more TCY

Only those differences that may impact the migration of an application are noted here.

Register and Bit Nomenclature

PIC18F and PIC24F devices all use a single control register and a single unlock register to control Flash operations. Some register and bit names, as shown in Table 7, change between the architectures.

TABLE 7: FLASH CONTROL REGISTERS AND BITS IN PIC18F AND PIC24F ARCHITECTURES

Flash Register or Bit Description	PIC18F	PIC24F
Flash Control Register	EECON1	NVMCON
Flash Key Register	EECON2	NVMKEY
EEPROM Data Register	EEPGD	—
PM/EEPROM Destination Select bit	CFGS	—
Flash Operation Enable bit	FREE	ERASE

The key Flash control bits (WR, WREN and WRERR) have the same names and functions in both architectures.

Flash Addressing

In PIC18F devices, the target address of Flash operations is specified by a 22-bit Table Pointer register, TBLPTR, that is composed of 3 single byte SFRs (TBLPTRL, TBLPTRH and TBLPTRU). All table read and write instructions use this pointer to address the entire memory space; however, only table read operations are allowed when the configuration space is addressed (TBLPTR<22> = 1).

In PIC24F devices, the target address for Flash operations is specified by two registers: the 8-bit Table Page register (TBLPAG) which specifies a 64 Kbyte area of the program memory space, and one of the W registers to specify an exact address within a designated page. The value of the two registers together forms the Effective Address or EA. As with PIC18F devices, only read operations are allowed when the configuration space is addressed (TBLPAG<8> = 1).

Data Size in Table Operations

Table operations in the PIC18F architecture are all byte-oriented. All bytes in the memory space are individually addressable, with the particular byte (high or low) defined by an odd or even address argument.

As already mentioned, table operations in the PIC24F architecture can be byte or word-oriented. In the latter, the specific byte (high or low) is defined by the instruction mnemonic itself (TBLRDH/TBLRDL and TBLWTH/TBLWTL). The byte option can also be specified to specifically operate on either byte of any given word. Keep in mind, of course, that operations on the upper byte of the upper word will either return 00h, or have no effect, for reasons previously described.

Reading Program Memory in Run Time

PIC18F devices can retrieve data from the program space using the table read command. Data is retrieved one byte at a time.

PIC24F devices can also retrieve data with table reads, either one word or one byte at a time. In addition, they can also use the PSV feature to map a 32 Kbyte block of the program space into the top half of the data space on a read-only basis. This is further explained in the **"Data Memory Space"** section on page 11.

DATA EEPROM

Many PIC18F devices offer on-chip, nonvolatile data memory for the storage of constant or slowly changing application data. These EEPROM areas are readable and writable, and are controlled by the same registers used to operate the Flash program memory.

As of this writing, PIC24F devices do not incorporate data EEPROMs in any devices. Users requiring a place to keep nonvolatile application data can use an unused block of program memory for this purpose. Data can be accessed and modified using table read and write operations, and read using PSV mode.

Data Memory Space

The PIC24F data memory space is substantially different than that found in PIC18F devices. The size, organization and method of access have all been changed. The key differences between the two architectures are shown in Table 8.

TABLE 8: COMPARISON OF MAJOR DATA MEMORY SPACE FEATURES

Feature	PIC18F	PIC24F
Addressing Range (size)	12 bits (4,096 byte maximum)	16 bits (65,536 byte maximum)
Segmentation	Linear range, banked addressing; linear addressing for some instructions	Linear range, no segmentation
Special Access Areas	Access RAM (bottom of lowest bank, top of highest bank)	Near Data Space (bottom 8K)
SFR Location	Top half of highest bank	Distributed throughout Near Memory
Stack	Hardware, 32 levels deep, not mapped in memory space	Soft stack starting at 0800h, user-configurable end of stack
Data Access	Byte (direct or indirect)	Double word, word or byte (all direct or indirect)
Hardware PSV	No	Yes, into top half of data space

ADDRESS RANGE AND SEGMENTATION

All PIC18F devices have a data memory space with a 12-bit address range. In theory, the data space has a linear range and can be addressed directly by several of the PIC18F instructions. For the most part, however, the data space functions as a segmented space. Since most PIC18F instructions can only contain the 8 lower bits of a data address, the data space is effectively divided into 16 banks of 256 bytes each. The exact memory location is also determined by the Bank Select Register (BSR) which contains the upper 4 bits of the address. The entire range of the data space is 4 Kbytes, of which some or all, may be implemented as data RAM.

In contrast, the PIC24F data space is implemented as a single linear range of addresses. Most instructions can directly access any address within the first 8 Kbytes of the range without the use of bank selection. The entire data space range is 64 Kbytes. Of this, only the first 32 Kbytes are implementable as data RAM; the upper 32 Kbytes are a virtual memory space that is used for PSV (see **"Program Space Visibility (PSV)"** on page 11).

A comparison of the data space maps is shown in Figure 4.

SFR LOCATIONS

In PIC18F architecture, all SFRs are located at the very top of data memory, generally from addresses F60h to FFh, as a more or less contiguous block.

In PIC24F architecture, SFRs reside in the lowest 2 Kbytes of the memory space, from addresses 0000h through 07FFh.

SPECIAL ACCESS AREAS

The effective segmentation of the PIC18F data space makes it necessary for some way of accessing SFRs and critical application data quickly. This is done by creating a virtual data space bank, known as the Access RAM, which is composed of the lower half of the lowest bank and the upper half of the upper bank. This scheme makes certain that the SFR space is always available, regardless of the contents of the BSR. Use of the Access RAM is included as an argument in PIC18F assembly language and is hard-coded in the instruction's opcode.

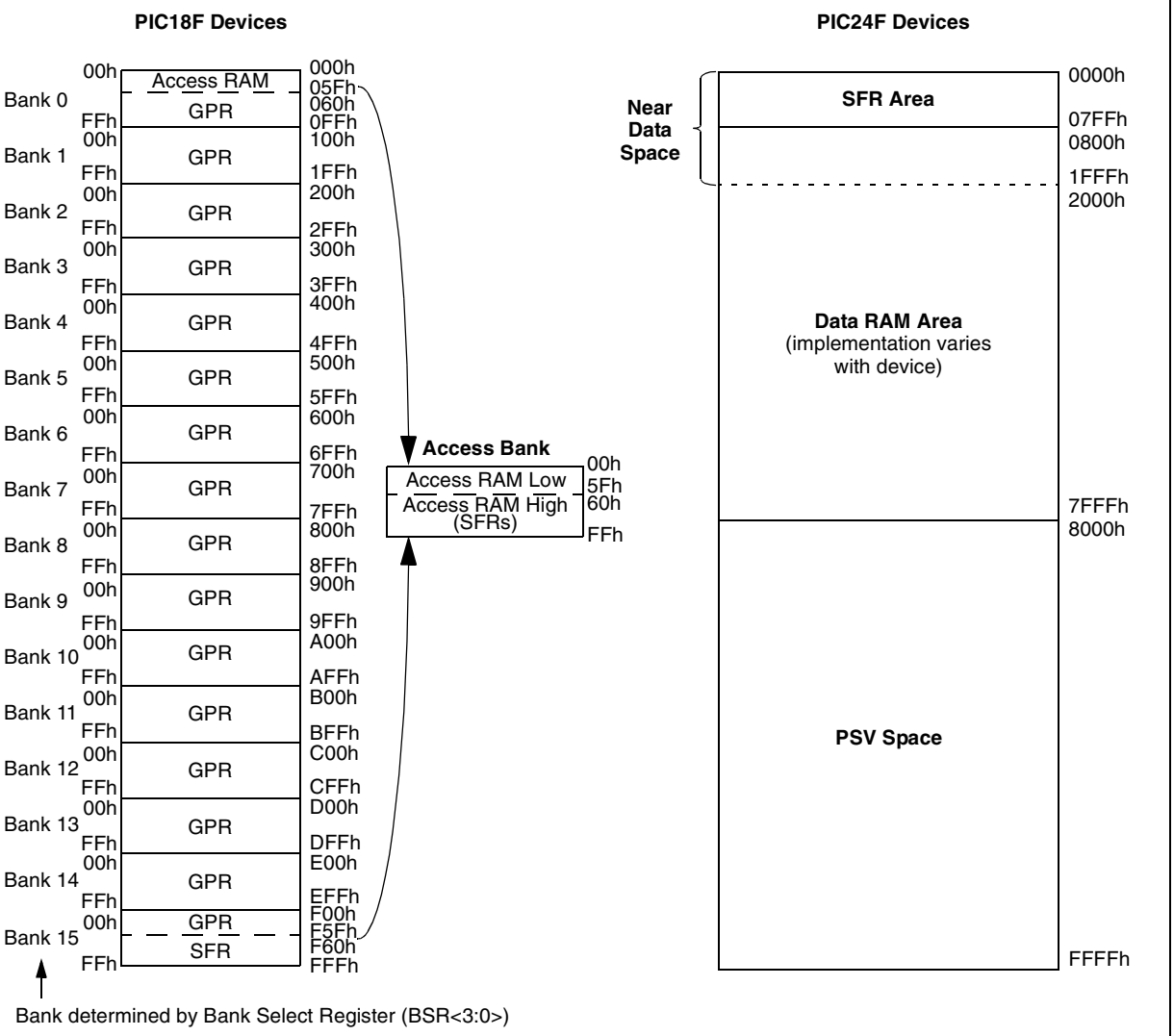
In the PIC24F data space, the first 8 Kbytes of data RAM between the addresses of 0000h and 1FFFh are referred to as the Near Data Space. Addresses in this space, including all SFRs, are accessible directly from all direct memory access instructions.

PROGRAM SPACE VISIBILITY (PSV)

Both PIC18F and PIC24F architectures allow for the direct access of information stored in the program memory space as data. For PIC18F, data from program memory is read in the data space by use of `TBLRD` commands, with access being done on a word-by-word basis.

For PIC24F devices, program memory is also made available through hardware-enabled Program Space Visibility (PSV). When used, any 32 Kbyte segment of the program space may be mapped into the upper 32 Kbyte area of the data space on a read-only basis. PSV uses a hardware register, PSVPAG, to define which page of program memory will be mapped. The PSV is controlled in software by the PSV bit (CORCON<2>).

FIGURE 4: COMPARISON OF PIC18F AND PIC24F DATA SPACE MAPS



Note: GPR = General Purpose RAM. Memory spaces are shown at different relative scales.

PROGRAM STACK

As discussed in the **"CPU Core"** on page 2, PIC18F devices use a hardware stack for program flow management. The stack is not memory mapped and has a fixed size of 32 levels.

PIC24F architecture uses a stack implemented entirely in mapped data space. The stack begins at 0800h in Near Data Memory, just outside of the SFR area, and grows towards higher memory addresses, using the W15 register as a dedicated pointer. The size of the stack is entirely user-defined with the SFR register, SPLIM, which sets the address for stack overflow traps.

DATA ACCESS

As discussed in the **"Instruction Set"** section on page 4, PIC18F architecture can only work with data in terms of bytes. In contrast, the PIC24F data space, organized in 2-byte words, allows many instructions to work with data as bytes, words or double words (32 bytes). The data type is determined by the argument used with the instruction.

Resets and Start-up Timing

The PIC24F device Reset system can be thought of as a superset of the PIC18F version. The same legacy Resets are supported in either identical or functionally equivalent methods:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- External Master Clear Reset (MCLR)
- Software Reset (RESET instruction)
- Watchdog Timer (WDT) Reset
- Stack Error (Overflow or Underflow)

PIC24F devices enhance the system with Resets for additional error states, along with enhanced reporting. The Reset states for SFRs and start-up timing from Resets also differs slightly. The major differences are summarized in Table 9.

TABLE 9: COMPARISON BETWEEN PIC18F AND PIC24F RESETS

Feature Description	PIC18F	PIC24F
Legacy Reset Types	POR, BOR, $\overline{\text{MCLR}}$, RESET Instruction, WDT, Stack Error	
Additional Reset Types	Configuration Word Mismatch (PIC18FXXJ Flash devices)	Illegal Opcode/Uninitialized W, Configuration Word Mismatch, Trap Conflict
BOR Configuration	Configurable, software controllable in many devices	Tied to on-chip regulator
Stack Underflow/Overflow Reset	Reset	Unmaskable trap
SFR Reset States(1)	Dependent on type of Reset	Uniform for all Reset types
Start-up Timer	Configurable	Tied to regulator configuration

Note 1: Excluding RCON and OSCCON registers.

ENHANCED EVENT REPORTING

All PIC18F devices use a total of 7 flag bits to report the most immediately past Reset event. This includes 5 bits in the RCON register and 2 stack event flags in the STKPTR register. PIC18F devices, with later implementations of nanoWatt Technology, also include a control bit (SBOREN), the Reset state of which can be used as an indirect flag. In all PIC18F devices, RCON flag bits are inverted sense: they are set in the absence of the event and cleared when the event has occurred.

The overlap of some bits in the indication of a particular Reset state requires that all bits be assessed to determine which Reset event has just occurred.

In contrast, PIC24F devices use 12 flag bits; 11 are located in the RCON register and one (STKERR) in the INTCON1 register. Each bit is closely associated with a particular event, so the change of state in one bit gives a clearer idea of the event that has just occurred. In addition, separate Sleep and Idle flag bits are now used to indicate if the device is in one of the Power-Saving modes. These can be used to identify exits on interrupt from Sleep or Idle modes. All flag bits are standard sense and become set when the corresponding event occurs.

The differences in Reset flag bits between PIC18F and PIC24F are summarized in Table 10. The specific bit states and their corresponding Reset states are compared in Table 11 and Table 12.

TABLE 10: COMPARISON OF RESET FLAGS IN PIC18F AND PIC24F ARCHITECTURES

RCON Bit Function	PIC18F	PIC24F
$\overline{\text{MCLR}}$ Reset	—	EXTR
Software Reset	$\overline{\text{RI}}$	SWR
POR	$\overline{\text{POR}}$	POR
BOR	$\overline{\text{BOR}}$	BOR
WDT Reset	$\overline{\text{TO}}$	WDTO
Trap Conflict Reset	—	TRAPR
Illegal Operation or Uninitialized W Register	—	IOPUWR
Configuration Word Mismatch	—	CM
Sleep Mode	$\overline{\text{PD}}$	SLEEP
Idle Mode	$\overline{\text{PD}}$	IDLE
Stack Overflow	STKFUL	STKERR(1)
Stack Underflow	STKUNF	STKERR(1)

Note 1: Implemented as a hardware trap rather than a Reset. See the **"Interrupt Controller"** section on page 17 for more details.

TABLE 11: PIC24F RESET STATUS BITS STATES

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	SLEEP	IDLE	CM	BOR	POR	STKERR
Power-on Reset	000000h	0	0	0	0	0	0	0	u	0	1	0
RESET Instruction	000000h	0	0	0	1	0	0	0	u	0	0	0
Brown-out Reset	000000h	0	0	0	0	0	0	0	u	1	0	0
$\overline{MCLR}$: in Run mode	000000h	0	0	1	0	0	0	0	u	0	0	0
in Idle mode	000000h	0	0	1	0	0	0	1	u	0	0	0
in Sleep mode	000000h	0	0	1	0	0	1	0	u	0	0	0
WDT Time-out Reset: in Run mode	000000h	0	0	0	0	1	0	0	u	0	0	0
in Idle mode	PC + 2	0	0	0	0	1	0	1	u	0	0	0
in Sleep mode	PC + 2	0	0	0	0	1	1	0	u	0	0	0
Stack Overflow Reset	000000h	0	0	0	0	0	0	0	u	0	0	1
Stack Underflow Reset(1)	000000h	0	0	0	0	0	0	0	u	0	0	1
Trap Conflict Reset(1)	000000h	1	0	0	0	0	0	0	u	0	0	0
Illegal Opcode/Uninitialized WREG	000000h	0	1	0	0	0	0	0	u	0	0	0
Configuration Word Mismatch Reset	000000h	u	u	u	u	u	u	u	1	u	u	u
Interrupt Exit: from Idle mode	(2)	0	0	0	0	0	0	1	u	0	0	0
from Sleep mode	(2)	0	0	0	0	0	1	0	u	0	0	0
Idle mode (execute PWRSAV 1)	PC + 2	0	0	0	0	0	0	1	u	0	0	0
Sleep mode (execute PWRSAV 0)	PC + 2	0	0	0	0	0	1	0	u	0	0	0

Legend: u = unchanged

Note 1: PIC24F stack events are trap events and not Resets; they are listed here for comparison purposes only. See the **"Non-Maskable Traps"** section on page 17 for more information.

2: Program counter is loaded with PC + 2 if the interrupt's priority is less than or equal to the CPU interrupt priority level, or the interrupt's hardware vector if the priority is greater than the CPU priority.

TABLE 12: PIC18F COMPARABLE RESET STATUS BITS STATES

Condition	Program Counter	RCON						STKPTR	
		SBOREN(1)	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	STKFUL	STKUNF
POR	0000h	1	1	1	1	0	0	0	0
BOR	0000h	u(2)	1	1	1	u	0	u	u
RESET Instruction	0000h	u(2)	0	u	u	u	u	u	u
$\overline{MCLR}$: in PRI_RUN mode	0000h	u(2)	u	u	u	u	u	u	u
in SEC_RUN or RC_RUN modes	0000h	u(2)	u	1	u	u	u	u	u
in Idle or Sleep modes	0000h	u(2)	u	1	0	u	u	u	u
WDT Time-out: in any Run mode	0000h	u(2)	u	0	u	u	u	u	u
in Idle or Sleep modes	PC + 2	u(2)	u	0	0	u	u	u	u
Stack Full Reset (STVREN = 1)	0000h	u(2)	u	u	u	u	u	1	u
Stack Underflow Reset or Error (STVREN = 1 or 0)	0000h	u(2)	u	u	u	u	u	u	1
Interrupt Exit from Power-Managed modes	PC + 2(3)	u(2)	u	u	0	u	u	u	u

Legend: u = unchanged

Note 1: SBOREN is not implemented in PIC18FXXJ Flash devices, or in devices with earlier versions of nanoWatt Technology.

2: Reset state is '1' for Power-on Reset and unchanged for all other Resets when software Brown-out Reset is enabled (BOREN1:BOREN0 Configuration bits = 01 and SBOREN = 1); otherwise, the Reset state is '0'.

3: When the wake-up is due to an interrupt and the GIEH or GIEL bit is set, the PC is loaded with the interrupt vector (008h or 0018h).

ADDITIONAL PIC24F RESETS

Beyond the legacy Reset events supported by both architectures, PIC24F supports three new hardware Resets:

- IOPUWR: Attempted execution of an unrecognized or undetectable instruction opcode, or an attempt to use an uninitialized register as an Address Pointer
- CM: An event has caused a mismatch between the Flash Configuration Words and their counterparts in configuration space, or the device has detected a parity error in the Configuration Words
- TRAPR: Two trap events have occurred, with the second taking place before the first has been cleared

These events have no equivalent in the PIC18F architecture.

STACK ERROR EVENTS

In PIC18F devices, a stack overflow or underflow event causes a full device Reset when the STVREN Configuration bit is programmed. Even if a stack interrupt is not enabled, the PC is reset to zero if an event occurs. The STKFUL or STKUNF flag bit is set in either event to indicate which event has occurred.

In PIC24F devices, a stack event is treated as a soft trap and causes code execution to jump to the interrupt vector at address 000Ah. No differentiation is made between overflow and underflow events. PIC24F trap events, in general, are discussed in more detail in the **"Non-Maskable Traps"** section on page 17.

BOR CONFIGURATION

PIC18F devices have always provided some level of configurability for BOR. At a minimum, the BOR can be enabled or disabled, and the BOR threshold (V_{BOR}) set during device programming, by using the BOREN and BORV1:BORV0 Configuration bits. Later PIC18F devices with nanoWatt Technology allow the BOR to be selectively controlled in software.

For PIC24F devices, the BOR function is incorporated into the on-chip voltage regulator. The BOR is always enabled when the regulator is enabled, and active while V_{DD} is below V_{BOR} (2.35V to 2.6V, typically). The ability of PIC24F devices to operate at maximum frequency across the normal operating voltage range makes a configurable BOR unnecessary. This is in contrast to many PIC18F devices, where maximum frequency may be tied to V_{DD}, as described in the specific device data sheet. In these cases, it may be necessary to configure a BOR set point that reflects the application's specific voltage and frequency requirements.

Note:	PIC18FXXJ Flash devices, such as the PIC18F87J10, use BOR circuitry identical to PIC24F devices.

SFR RESET STATES

In PIC18F devices, SFRs have up two designated Reset states: the default state on POR or BOR, and an alternate state on all other legacy Resets. Depending on the register's function, the register values in the two states are identical; but for many registers, the value prior to the Reset is preserved.

In PIC24F devices, there is no distinction between POR/BOR and other Resets. All registers return to their original default values on a device Reset, regardless of the source. The only exceptions to this rule are the OSCCON and RCON registers, where the Reset value of one or more bits is dependent on device configuration or state immediately prior to Reset.

Data sheets and other literature for PIC18F devices often list a third Reset state: the value immediately following a wake-up by WDT time-out or interrupt. For almost all registers in all PIC18F devices, the value is the same as that immediately before the wake-up event. For PIC24F devices, all SFRs are assumed to remain unchanged following a wake-up. Specific exceptions will be noted in the appropriate device data sheet.

START-UP TIMING AFTER RESETS

PIC18F devices include a fixed start-up timer that holds the microcontroller in a Reset state. The timer's interval, T_{PWRT}, allows for the memory array to power-up and internal circuits to reach their initial states. The power-up timer can be enabled or disabled during device configuration by programming the PWRT Configuration bit. Additional delays are added to T_{PWRT} depending on the start-up oscillator selected and whether or not the internal PLL is being used.

For PIC24F devices, the start-up time is dependent on the configuration of the on-chip voltage regulator when the device exits a POR or BOR event. The total regulator delay is $T_{STARTUP}$, which is equal to T_{PWRT} (if the regulator is disabled) or T_{VREG} (if it is enabled). T_{PWRT} is a fixed, 64 ms nominal delay and is comparable to the PIC18F T_{PWRT} delay; it provides time for the power supply to stabilize. T_{VREG} is a nominal, 10 µs delay which permits the internal voltage regulator to stabilize. An additional voltage regulator delay occurs if VREGS (RCON<8>) is clear. Additional delays are also provided for oscillator start-up and PLL, if enabled, which are consistent with PIC18F devices. See the device data sheets for more information.

OTHER RESET FEATURES

Many PIC18F devices with later implementations of nanoWatt Technology have a configurable $\overline{MCLR}$ pin. For these devices, the MCLRE Configuration bit can select if the pin functions as the device Master Clear, or as an input only I/O port. PIC24F devices have "fixed" $\overline{MCLR}$ pins that cannot be reconfigured.

MIGRATION CONSIDERATIONS

For applications that require polling of Reset events, the method of polling will need to be changed. Although all states that were tracked in PIC18F devices are still trackable in PIC24F, the meaning of the legacy RCON flag bits is more narrowly defined. It is no longer necessary to read all flags to interpret the immediately previous event; for most cases, it will only be necessary to see which flag bit is now set. Note that the PIC24F RCON status bits are active-high and PIC18F RCON bits are active-low.

Be aware that the start-up time of the device increases when the regulator is disabled to permit the application voltage to stabilize. Enabling the regulator also enables the BOR function.

Note:	The internal regulator should not be used to power circuitry outside of the device. It is intended for internal power consumption only.

It is recommended for both architectures to connect $\overline{MCLR}$ to V_{DD} using a 1 kΩ resistor. This will limit the current flow into the $\overline{MCLR}$ pin due to potential external overstresses.

Interrupt Controller

The PIC24F interrupt controller contains many improved interrupt options compared to the PIC18F. These new features include, but are not limited to, increased processor exceptions, software traps and user-selectable priority levels. Table 13 summarizes the differences between the PIC18F and PIC24F interrupts.

TABLE 13: COMPARISON BETWEEN PIC18F AND PIC24F INTERRUPT CONTROLLERS

Interrupt Features	PIC18F	PIC24F
Unique Interrupt Flag and Enable for each Source	Yes	Yes
Global Interrupt Enable	Yes	Yes
Software Clearable Interrupt Flags	Yes	Yes
Flags Set Regardless if Interrupt is Enabled or Disabled	Yes	Yes
Software can Generate any Peripheral Interrupt	Yes	Yes
Automatic Context Save/Restore	Yes	Yes
Exits Power-Managed Modes	Yes	Yes
Assignable Interrupt Priority	high or low	8 levels, user-defined
Latency for External Interrupt Events (INTx pins or the PORTB input change interrupt)	3 or 4 TCY	5 TCY (Fixed)
Priority Exit from Sleep and Idle Modes	No	Yes
Interrupt Nesting Disable Option	No	Yes
Software Selectable Core Interrupt Priority Level	No	Yes
Trap Vectors	No	Yes (4)
Unique Interrupt/Trap Source	No	Yes
Alternate Interrupt Vector Table for Emulation	No	Yes
Natural priority Unmaskable or Non-Maskable Interrupts	No	Yes
Capable of Disabling Interrupts for Specific Number of TCY	No	Yes

NEW PIC24F INTERRUPT FEATURES

- **Unique Interrupt and Trap Vectors:** Each individual interrupt source is assigned in hardware a natural priority, allowing interrupts to be sorted with no additional user intervention, as well as a unique interrupt vector
- **User-Assignable Priority:** Users can also give each interrupt one of eight levels of priority, which can be used to override the natural priority
- **Software Assigned Core Priority:** Users can also set a threshold priority level at which the CPU will respond to interrupts
- **Interrupt Nesting:** The use of natural priority and user-assigned priority allows multiple interrupt events to be nested; this feature can also be selectively disabled
- **Hard and Soft Traps:** Up to 8 non-maskable hard traps with high natural priority are provided to flag potentially serious events, such as math (divide by 0), stack overflow/underflow, address or data alignment and oscillator failure
- **Alternate Vector Table:** Allows for a convenient switch between support and application environments without reprogramming
- **Priority Exit From Power-Saving modes:** Allows the application to either resume normal code execution, or jump to an ISR, depending on the interrupt priority level

Note: For further information on traps, refer to the specific device data sheet.

UNSUPPORTED PIC18F FEATURES

All PIC18F interrupt controller features are supported in the PIC24F interrupt controller.

NON-MASKABLE TRAPS

In PIC24F architecture, there are four hardware trap events with interrupts that can never be disabled:

- Address Decode Error
- Oscillator Failure
- Stack Error
- Math (Overflow) Error

These errors always force an immediate jump to specific interrupt vectors. The two most serious errors (Address Decode and Oscillator Failure) are hard traps; these must be cleared before the CPU execution can continue. All traps have their own individual flag bit.

In addition to these unmaskable events, the PIC24F architecture can be expanded at a future time to include up to four additional traps.

PIC18F architecture does not have an equivalent to the hardware trap. PIC18F stack error events are treated as Resets.

BIT NAME CHANGES AND MAPPING

PIC24F devices maintain the same general nomenclature for interrupt bit names as PIC18F devices, with two important differences. Both families maintain interrupt enable, flag and priority bits that are generically named xxxIE, xxxIF and xxxIP (where 'xxx' is the mnemonic for the interrupt source).

The first major difference is the presence of three interrupt priority bits for each source, instead of the one used for PIC18F devices. These bits, generically named xxxIP2 through xxxIP0, allow the interrupt to be assigned one of eight relative priority levels.

The other difference is the number of interrupt sources. While many interrupts have the same (or very similar) name as PIC18F devices, others are new. Other interrupts have similar names but have a different meaning from their PIC18F counterparts. Users should refer to the appropriate PIC24F device data sheet for a complete list of interrupts and their meanings.

SETUP AND ENABLING INTERRUPTS

To set up and enable interrupts on PIC18F devices:

Required Steps:

1. Clear the interrupt flag status bit associated with the peripheral in the associated PIRx or INTCONx register.
2. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate PIEx or INTCONx register.

Note:	The interrupt flag still needs to be cleared prior to exiting an ISR.

Optional Step:

Select the user-assigned priority level for the interrupt source by writing to the control bits in the RCON register; select high priority or low priority using the priority bit in the corresponding IPRx register. The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>).

To set up and enable interrupts on PIC24F devices:

Required Steps:

1. Set the NSTDIS bit (INTCON1<15>) if nested interrupts are not desired.
2. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
3. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

Note:	The Interrupt/trap flag still needs to be cleared prior to exiting an ISR.

Optional Step:

Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bit for all enabled interrupt sources may be programmed to the same non-zero value.

Note:	Upon Reset, all interrupts are assigned a default priority level of 4.

DISABLING USER INTERRUPTS

To disable interrupts on PIC18F devices, it is only necessary to clear the GIE bit (GIEH or GIEL if priority levels are used).

To disable user interrupts on PIC24F devices, these steps are required:

1. Push the current STATUS register value onto the software stack using the `PUSH` instruction.
2. Force the CPU to priority level 7 by inclusive ORing the value OEh with the low byte of the STATUS register (SRL). To enable user interrupts, the `POP` instruction may be used to restore the previous STATUS register value.

Note:	The `DISI` instruction allows interrupts of priority levels 1-6 to be disabled for a fixed period of time.

MIGRATION CONSIDERATIONS

PIC18F architecture only has the ability to assign either high or low priority interrupts to individual sources. PIC24F architecture allows the assignment of multiple priority levels for interrupts (priorities 0 through 7 are user defined and priorities 8 through 15 are hardware defined). At the very least, interrupts in native PIC18F applications will need to be re-assessed, and their priority levels redefined in PIC24F terms.

For both PIC18F and PIC24F devices, the `RETFIE` instruction exits an Interrupt Service Routine (ISR), but this instruction does behave slightly different depending on the microcontroller. For PIC18F, this instruction will set the GIE bit to re-enable global interrupts. Since the GIE bit does not exist for PIC24F, this instruction will restore the previous priority level.

Oscillator

The PIC24F oscillator system supports many of the features of PIC18F nanoWatt Technology, and adds several new features. Both architectures support three major clock sources: primary oscillators, internal RC oscillators and 4x PLL frequency multipliers. In addition, both also support features to enhance application robustness, such as software-controlled clock switching, Fail-Safe Clock Monitors and Two-Speed Start-up. PIC24F devices increase the flexibility of initial clock configuration, software controlled clock switching and the use of the PLL.

Although there are many similarities between PIC24F and PIC18F architectures in the oscillator and clock generation, there are also important differences in the implementation of these features. Because the oscillator affects most aspects of an application, it is important to understand the differences before migrating an application to a PIC24F device.

The major differences between the PIC18F and PIC24F clock structures are detailed in Table 14.

CHANGES FROM PIC18F ARCHITECTURE

Features added from the PIC18F implementation of nanoWatt Technology include:

- **Enhanced PLL Operation:** PIC24F architecture allows the PLL to be used under software control in all modes where it is available, and makes the PLL available to all External Primary Oscillator modes.
- **Doze Mode:** This permits differential clocking of the CPU and peripherals, and allows the CPU to run at lower speeds while maintaining timing sensitive peripherals (see the **"Power-Saving Features"** section on page 23 for more details).

PIC18F features that are no longer supported in PIC24F architecture include:

- LP and External RC Primary Oscillator modes have been removed in favor of SOSC and FRC Oscillator modes
- XT mode is still implemented, but is limited to the upper end of the upper range of frequencies available with PIC18F devices. For PIC24F devices, the XT frequency range is 3.5 MHz to 10 MHz.

TABLE 14: COMPARISON OF MAJOR OSCILLATOR FEATURES

Feature Description	PIC18F	PIC24F
Primary (External) Oscillator Modes	HS, XT, EC, LP and external RC(1)	HS, XT and EC (all devices)
Secondary (Timer1) Oscillator	Yes	Yes (SOSC)
8 MHz Internal RC Oscillator	Most devices (INTOSC)(1)	Yes (FRC)
31 kHz Internal RC Oscillator	Yes (INTRC)	Yes (LPRC)
4x PLL Options: XTPLL (MSPLL) ECPLL INTOSCPLL/FRCPLL	 No Select devices only Select devices only	 Yes Yes Yes
INTOSC/FRC Tuning	5-bit magnitude, ±12% range	6-bit magnitude, ±12% range
Software Clock Switching	Between clock sources only	Between all available oscillator types; uses safety interlock
Doze Mode	No	Yes
Fail-Safe Clock Monitor	Yes	Yes
Two-Speed Start-up	Yes	Yes

Note 1: Applies to most PIC18FXXXX devices, such as PIC18F8722. Currently, PIC18FXXJ Flash devices, such as the PIC18F87J10, only offer HS and EC Primary Oscillator modes and no INTOSC oscillator.

BIT AND CONTROL FUNCTION MAPPING

There is not a precise one-to-one correspondence between configuration and control functions of the PIC18F and PIC24F architectures. The most common differences are shown in Table 15. Details are discussed later in this section.

TABLE 15: COMPARISON OF OSCILLATOR BIT AND CONTROL FUNCTION MAPPING

Oscillator Block Function	PIC18F	PIC24F
Define Start-up Oscillator	FOSC3:FOSC0 (CONFIG1H<3:0>)(1)	FNOSC2:FNOSC0 (Configuration Word 2<10:8>)
Define Primary Oscillator Type		POSCMD1:POSCMD0 (Configuration Word 2<1:0>)
Define I/O Port Functions in EC Oscillator Modes		OSCIOFCN (Configuration Word 2<5>)
Enable Secondary Oscillator	T1OSCEN (T1CON<3>)	SOSCEN (OSCCON<1>)
Select INTOSC/FRC Postscaler	IRCF2:IRCF0 (OSCCON<6:4>)	RCDIV2:RCDIV0 (CLKDIV<10:8>)
Switch Run-Time Clock Source	SCS1:SCS0 (OSCCON<1:0>)	NOSC2:NOSC0, OSWEN (OSCCON<10:8, 0>), FCKSM0 (Configuration Word 2<6>)
Active Clock Source Monitoring	OSTS, IOFS, T1RUN	COSC2:COSC0
Enable Two-Speed Start-up	IESO (CONFIG1H<7>)	IESO (Configuration Word 2<15>)
Enable Fail-Safe Clock Monitor	FCMEN (CONFIG1H<6>)	FCKSM1:FCKSM0 (Configuration Word 2<7:6>)

Note 1: PIC18FXXXX devices only; FOSC2:FOSC0 (CONFIG2H<2:0>) on current PIC18FXXJ Flash devices.

PRIMARY OSCILLATORS (POSC)

PIC18F microcontrollers, with the latest versions of nanoWatt Technology, use an oscillator circuit that supports a wide range of external components, including a variety of crystals, RC networks or external clock generators. To use the oscillator, users connect their crystal or RC circuit to the OSC1 and OSC2 pins, or present the external clock on OSC1. The exact oscillator mode to be used is selected during device configuration using the FOSC3:FOSC0 Configuration bits, and must match the actual external circuit to be used.

PIC24F microcontrollers support a similar range of oscillator options on the OSC1 and OSC2 pins; however, the options for support of low-power, low-frequency crystals and external RC resonant circuits have been omitted. These have been replaced by the use of a high-speed internal RC oscillator, and the use of the Timer1 oscillator in the more flexible clock structure.

The primary oscillator mode is selected during configuration with a combination of the FNOSC2:FNOSC0 and POSCMD<1:0> Configuration bits. Like PIC18F devices, the configuration must match the external circuit to be used in the application.

SECONDARY OSCILLATOR (SOSC)

All PIC18F devices have the option to use the Timer1 oscillator as a secondary clock source. The most typical arrangement for this option is to connect a low-power, 32 kHz watch crystal across pins T1OSI and T1OSO. The oscillator is controlled separately from the device clock controls with the T1OSCEN bit (T1CON<3>).

PIC24F devices also provide a secondary oscillator that is identical in function to the Timer1 oscillator; it only differs in that it is controlled through the OSCCON register with the SOSCEN bit. The crystal input/output pins are renamed SOSCI and SOSCO.

The PIC24F secondary oscillator takes the place of the Primary LP Oscillator mode provided on PIC18F devices, but not supported on PIC24F. To use SOSC as the default oscillator on start-up, it must by selected in device configuration by the FNOSC2:FNOSC0 bits and enabled in software by setting the SOSCEN bit.

INTERNAL RC OSCILLATORS (INTOSC/FRC AND INTRC/LPRC)

The internal oscillators for PIC18F devices with nanoWatt Technology and PIC24F devices are virtually identical, except in name. Both feature two independent internal oscillators, an efficient 31 kHz oscillator and an accurate, high-speed 8 MHz oscillator. Both architectures use a configurable postscaler, driven by the 8 MHz source, to provide a range of clock frequencies, from 31 kHz to 4 MHz (as well as the undivided 8 MHz output). Both architectures allow software selection from the 31 kHz or 8 MHz oscillators to provide the 31 kHz source for various system features. Both clock systems permit tuning of the 8 MHz source through a nominal range of ±12%.

The differences here are minor. For PIC18F devices, the internal 31 kHz and 8 MHz sources are generally referred to as INTOSC and INTRC. For PIC24F devices, they are known as the Fast RC (FRC) and Low-Power RC (LPRC) oscillators. INTOSC/FRC tuning is accomplished with 5 tuning bits (TUN4:TUN0) in PIC18F devices; PIC24F devices use 6 tuning bits (TUN5:TUN0) for finer resolution. Finally, the default value on Reset for the 8 MHz postscaler is 1 MHz for PIC18F devices and 4 MHz for PIC24F devices.

PLL FREQUENCY MULTIPLIER

PIC18F and PIC24F devices both support a 4x PLL frequency multiplier for use with select clock sources. In all cases, the PLL provides a stable output only when the input frequency is between 4 and 10 MHz. The operation of the PLL differs substantially between the two architectures.

For most PIC18F devices incorporating nanoWatt Technology, the PLL is automatically enabled for specific, primary oscillator configurations and is always operational. All devices can use the HS oscillator with the PLL; select later devices also allow the use of the PLL with the EC mode. These are distinct primary oscillator configurations. If the user wishes to disable the PLL, the device must be reprogrammed and reconfigured. If the internal oscillator block is selected as the device's default oscillator, the PLL is made available when the INTOSC postscaler is configured for an output of 4 or 8 MHz. In these cases, the PLL can be selectively enabled under software control with the PLLEN control bit (OSCTUN<6>).

For PIC24F devices, the PLL is always available under software control. It is available for all primary oscillator modes, as well as FRC or FRCDIV operation (as long as a postscaler output of at least 4 MHz is selected). To use the PLL, it is only necessary to perform a clock switch to one of the PLL Clock modes. Once a PLL mode is selected, the state of the PLL's output stability is indicated by the flag bit, LOCK (OSCCON<5>). When the bit is set, the PLL output is stable.

CLOCK SWITCHING

Clock switching differs significantly between PIC18F and PIC24F devices. Conceptually, both architectures have three categories of oscillators: primary (external components connected to OSC pins), secondary (external crystal connected to T1OSC or SOSC pins) and internal RC. PIC18F devices permit the definition of one and only one primary oscillator type used during device configuration. This is the oscillator that is always used when on device power-up and Reset. Thereafter, the device can switch between primary, secondary and internal oscillator sources, under software control, by writing to the SCS1:SCS0 bits. Once a primary oscillator is defined, it cannot be changed unless the device is reprogrammed.

For PIC24F devices, any one of the three major clock sources can be configured as the default start-up oscillator; users are no longer confined to just the primary oscillator sources. During run time, the device can switch between any of the available oscillator modes under software control. This means, that among other things, it is possible to switch between a Primary Clock mode and its PLL counterpart while the application is running. It is also possible to start the device using the Timer1 or LPRC oscillator, rather than switch to those sources after Reset or power-up (as was required in PIC18F implementations of nanoWatt Technology). In fact, it is possible to completely disable the primary oscillator source in PIC24F devices; something that cannot be done on the PIC18F architecture.

This increased flexibility makes clock switching on PIC24F devices a more complex sequence. The new oscillator is selected with the NOSC2:NOSC0 bits and by setting the Oscillator Switch Enable bit, OSWEN. To prevent unintended changes, PIC24F devices also use an additional safety interlock that requires an unlock sequence to write each byte of the OSCCON register. PIC24F unlocks the high or low byte for one instruction after two specific literals are written to the high or low byte of OSCCON. An instruction counter ensures the unlock sequence is performed within a maximum number of instructions and remains unlocked for one instruction cycle. Because these sequences are so time critical, the unlock sequences are done with an assembly language routine. When the NOSC bits match the COSC bits, or the OSWEN bit is clear, the clock switch has been completed successfully. Because the New Oscillator Select bits, NOSC, and Oscillator Switch Enable bit, OSWEN (OSCCON<0>), reside in opposite halves of OSCCON, two unlock sequences are needed to request a system clock switch. For examples of the OSCCON unlock sequences, refer to the specific device data sheet.

TWO-SPEED START-UP

Two-Speed Start-up is implemented identically in PIC18F and PIC24F devices. In both cases, the feature is controlled by the IESO Configuration bit.

FAIL-SAFE CLOCK MONITOR

The Fail-Safe Clock Monitor feature is also available for the PIC24F. It is controlled, along with run-time clock switching, by the Configuration Word bits, FCKSM1:FCKSM0 (CW2<7:6>). FSCM in PIC24F devices is similar to the PIC18F implementation by automatically switching to the FRC when the primary oscillator stops.

The single significant difference is the secondary effects of an FSCM event. For PIC18F devices, a primary oscillator failure sets the OSCFIF interrupt flag bit, which can optionally generate a device interrupt. For PIC24F devices, an FSCM event sets both the CF flag (OSCCON<3>) and OSCFAIL (INTCON1<1>) bits, and generates an unmaskable hardware trap which then must be cleared.

MIGRATING A TYPICAL APPLICATION

- Most applications built on a PIC18F device with nanoWatt Technology will be able to use the same oscillator type and clock frequency when a PIC24F device is substituted. This is particularly true when the oscillator uses a crystal (HS mode or XT mode, between 3.5 and 10 MHz), an external clock generator or the internal RC oscillator block.
- PIC18F applications that use a 32 kHz crystal for the primary oscillator will need to use the secondary oscillator for their PIC24F equivalent versions. The crystal circuit will need to be moved to the T1OSCI/T1OSCO pins, and the Secondary Oscillator mode (SOSC) will need to be selected as the start-up clock source.
- PIC18F applications using an External RC Primary Oscillator mode, or an XT Oscillator mode running below 3.5 MHz, must now use the FRC oscillator as the default clock source. Configuration will need to be changed to select FRC as the default start-up clock source.
- If clock switching is used, the clock switch sequences will need to be added. These are described in the appropriate device data sheet.

MIGRATION CONSIDERATIONS

When migrating to a PIC24F microcontroller (or any microcontroller, for that matter), any application that is based on a crystal clock source should be re-evaluated for oscillator operation and stability. It is important to verify that the crystal performance is reliable across the voltage, temperature and process variations anticipated for the application.

For more information, refer to the application notes listed in the **"References"** section on page 45.

Power-Saving Features

PIC24F power-saving features are very similar to the Power-Saving modes offered in PIC18F nanoWatt Technology devices. Both architectures include run-time switching of system clock sources, Idle and Sleep modes, and hardware invoked exits through Resets and interrupts. PIC24F devices describe these features in a somewhat different manner, and support additional features for strategic reduction of power consumption.

A comparison of the power-saving features in both architectures is presented in Table 16.

Note:	For the sake of brevity, references to "PIC18F" throughout this section should be interpreted as meaning "PIC18F devices with nanoWatt Technology".

TABLE 16: COMPARISON BETWEEN PIC18F AND PIC24F POWER-SAVING FEATURES

Feature Description	PIC18F	PIC24F
Run-Time Clock Switching	Yes	Yes
Power-Saving Mode Invocation	Instruction and hardware bit setting	Instruction with argument
Idle Mode Selective Peripheral Idle	Yes No	Yes Yes
Sleep Mode	Yes	Yes
Doze Mode	No	Yes
PMD Option	No	Yes

RUN-TIME CLOCK SWITCHING

PIC18F and PIC24F devices have all the same types of system clock sources (primary, secondary and internal oscillator). In addition, Sleep and Idle modes are defined in the same manner. The difference between the devices is strictly terminology: PIC24F devices do not use the Power-Managed mode terminology created for nanoWatt Technology. Because oscillator mode switching in more expansive in PIC24F devices, the old PIC18F descriptions of Power-Managed modes (PRI_RUN, SEC_IDLE, etc.) are no longer used. However, completely equivalent modes are available in PIC24F devices; that is, using the NOSC2:NOSC0 bits to select the FRC oscillator as the clock source is equivalent to switching to RC_RUN mode in a PIC18F device.

In PIC18F devices, clock switching is accomplished by writing to the SCS1:SCS0 bits (OSCCON<1:0>). For PIC24F devices, clock switching is accomplished by writing to the NOSC2:NOSC0 bits, accompanied by a safety unlock procedure (see the **"Oscillator"** section on page 19 for more details). In addition, clock switching can be disabled entirely by setting Configuration bit, FCKSM1 (CW2<7>).

The internal transitions between clock sources are essentially the same in both architectures and are accompanied by similar delays.

SLEEP AND IDLE MODES

PIC18F devices use the `SLEEP` instruction to invoke their Power-Managed (x_IDLE and Sleep) modes. The actual mode invoked is determined by the IDLEN bit (OSCCON<7>); an Idle mode is entered when the bit is set or Sleep mode when the bit is cleared.

In PIC24F devices, the equivalent instruction is `PWRSAV`. The instruction is used with an argument (either symbolic or literal) to specify the desired mode. As before, Idle mode is invoked when the literal argument is '`1`', and Sleep mode when it is '`0`'.

SELECTIVE PERIPHERAL IDLING

In PIC18F devices, Idle mode is an all-or-nothing affair; all peripherals remain operational during an Idle mode.

In PIC24F devices, peripherals can be selectively disabled during Idle mode. Most peripheral modules have a control bit, xxxIDL (where 'xxx' represents the peripheral mnemonic), that determines if the peripheral continues operation when Idle mode is invoked. The IDL bits allow modules to be partially powered down, providing for additional incremental power savings.

EXITING POWER-SAVING MODES

Exiting a Power-Managed mode can be achieved by an enabled interrupt or by a Reset. A WDT Reset or interrupt exits the Power-Managed mode and returns the device to the previous clock source. All other methods switch the controller to the 8 MHz FRC internal oscillator, if Two-Speed Start-up is enabled, until the primary clock source is ready.

DOZE MODES

PIC24F devices feature an additional Doze mode option to reduce power consumption while allowing the application to continue running. Doze mode permits the CPU to run at a reduced clock rate while the peripherals are clocked at full speed. It is enabled by setting the DOZEN bit (CLKDIV<11>). The CPU peripheral clock ratio is selected using the DOZE bits (CLKDIV<14:12>), with a range of 1:128 (slowest) to 1:1 (full speed). An option to disable Doze mode on an interrupt is available through the Recover on Interrupt bit, ROI (CLKDIV<15>).

PIC18F devices have no equivalent feature.

PERIPHERAL MODULE DISABLE

In PIC24F devices, the PMD feature reduces peripheral current consumption to the absolute minimum possible. Peripherals which support PMD have an associated control bit in one of the PMD registers. When enabled, all clock sources are disconnected from the peripheral, and all SFR registers associated with it appear as unimplemented memory. This is a contrast to the Selective Peripheral Idle, where the module is partially powered down but remains accessible.

There is no equivalent feature in PIC18F architecture.

ADDITIONAL PIC24F FEATURES

The internal voltage regulator can be brought into a low-power mode by setting the VREGS bit (RCON<8>). In this mode, the voltage regulator supplies only enough power to maintain RAM states. If any peripherals are active, this mode cannot be used.

MIGRATING A TYPICAL SETUP

For straight migrations of a PIC18F application using Power-Saving modes, there is a tight correspondence between PIC18F and PIC24F instructions. Users will need to change references from SCS1:SCS0 to the equivalent mode using NOSC2:NOSC0 (plus the addition of the register unlocking sequence). Additionally, all `SLEEP` instructions will need to be changed to the appropriate `PWRSV n` equivalent, and references to IDLEN removed.

MIGRATION CONSIDERATIONS

- The voltage/frequency characteristics of the PIC18F and PIC24F architectures are significantly different and will affect clock switching decisions. The PIC24F voltage supply range (2.0V to 3.6V) is narrower than for PIC18F devices, such as the PIC18F8722. Applications that change clock frequencies, due to reduced power supply voltage, may not need to switch at all, since the PIC24F can operate at higher speeds across a wider range of voltage. Also note that PIC18FXXJ Flash devices, such as the PIC18F87J10, have V/F characteristics that are similar to the PIC24F family.
- The lower maximum V_{DD} of PIC24F devices may also impact the design of an application migrated from PIC18F devices. Refer to the appropriate product data sheets for more information on operating voltage range and Power-Managed modes.

Watchdog Timer (WDT)

The Watchdog Timer (WDT) module for the PIC24F is nearly identical to the PIC18F nanoWatt device implementation. Both use Configuration bits to enable the WDT and to select a time-out period. The time-out periods are the same and both feature a software enable option. Power-Managed modes can also be exited using the WDT for both devices. In addition, the PIC24F implementation has a new configurable prescaler and windowed WDT options available in the Configuration Word.

TABLE 17: COMPARISON BETWEEN PIC18F AND PIC24F WDT FEATURES

Feature Description	PIC18F	PIC24F
Configurable Time-out Period	Yes	Yes
Software Enable	Yes	Yes
Exit Power-Managed Modes	Yes	Yes
Time-out Range	1 ms to 131s	1 ms to 131s
Prescaler	No	Yes
Windowed WDT Option	No	Yes

COMMON FEATURES

The WDT modules for both PIC18F and PIC24F devices can be enabled during programming in the Configuration Word. The PIC24F is enabled or disabled by the FWDTEN (CW1<7>) Configuration bit. A time-out period is also selected in the Configuration Word with the WDTPS<3:0> bits (CW1<4:0>), ranging from 1 ms to 131s. The periods are the same for both devices within the accuracy limitations of the 32 kHz Low-Power Internal Oscillator (LPRC). Both implementations automatically enable the 32 kHz LPRC if it is not already enabled.

Sleep and Idle modes can be exited using the WDT. If a time-out occurs in one of these modes, code execution resumes at the next instruction in program memory. Also, the WDT can be enabled in software by setting the SWDTEN (RCON<5>) bit if the option is disabled in the Configuration Word. Both architectures clear the WDT counter in the following events:

- `CLRWDT` instruction
- `SLEEP` instruction
- Selecting a new internal oscillator frequency

NEW PIC24F FEATURES

The PIC24F WDT has a prescaler option in the Configuration Word to select either 1:128 or 1:32 using the FWPSA bit (CW1<4>). To match the PIC18F nanoWatt implementation, set the FWPSA to select the 1:128 option.

The windowed WDT feature is enabled by clearing the Configuration bit, WINDIS. For more information on the windowed WDT option, refer to the applicable product data sheet.

UNSUPPORTED PIC18F FEATURES

All PIC18F WDT features are supported in the PIC24F WDT.

MIGRATING A TYPICAL SETUP

Migrating PIC18F WDT to PIC24F is straightforward. Enable the WDT in the same manner, either by setting the FWDTEN Configuration bit, or in software, by setting the SWDTEN bit in the RCON register. Select the same postscaler setting used in the PIC18F configuration with the WDTPS<3:0> bits and set the FWPSA Configuration bit to choose the 1:128 prescaler setting. The PIC24F WDT can be used in the same manner as the PIC18F to exit Power-Managed modes and monitor for unexpected code execution.

MIGRATION CONSIDERATIONS

- The PIC24F WDT counter will be cleared on any clock source switch, whether it was caused from the Fail-Safe Clock Monitor, or by software. This may differ from the PIC18F implementation that typically clears the WDT on the conditions specified in the section, **"Common Features"**, on page 25.
- When a WDT time-out occurs, PIC18F devices clear the $\overline{TO}$ (RCON<3>) bit and PIC24F devices set the WDTO (RCON<4>) bit. On POR or BOR, the state of these bits is unknown and must be initialized to the opposite state to detect a WDT time-out event.

Device Integration Features

Certain features of PIC18F and PIC24F devices are not considered as individual modules, but part of a system integration package that makes the devices more reliable and their applications more cost-effective. In addition to the WDT, included in this group are:

- Device Configuration Control
- Code Protection
- On-Chip Voltage Regulator

DEVICE CONFIGURATION

All PICmicro microcontrollers program their basic and generally, application constant configuration setting into an area of nonvolatile memory inside the device configuration space. The manner in which this information is stored and used differs greatly between PIC18F and PIC24F families.

All PIC18F devices use Configuration registers at addresses 300000h through 30000Fh, well outside of the normal program space. While all devices can read these addresses during run time, most cannot change their configuration settings through self-reprogramming. Changes to the Configuration Words can only be done with external programming, such as ICSP. In some early devices, the configuration settings were implemented with one-way fuses, and were essentially permanent.

Later versions of PIC18F devices, such as PIC18FXXJ Flash devices, store their configuration settings in the last four words at the top of program memory. At POR, the information is latched into the actual registers (still at addresses 300000h through 30000Fh), which controls the device configuration. The actual registers in configuration space cannot be directly written to by the user, or programmed via ICSP. While the configuration data in program memory can be altered during run time using table write operations, the new configuration is not latched into the actual Configuration registers and does not take effect until a POR occurs. Even then, changes to device code protection cannot be done without a Chip Erase and complete reprogramming. The actual Configuration registers are parity-protected against spurious bit changes; any unexpected changes will result in a device Reset.

Like PIC18FXXJ Flash devices, PIC24F devices store their Configuration Words at the end of implemented program memory. Unlike PIC18F, these devices reread and latch the configuration settings on all types of device Resets, including the use of the `RESET` instruction. This means that any aspect of device configuration, except for code protection, can essentially be updated in software at any time. Code protection settings are still specially protected, and can only be changed by a Chip Erase and complete reprogramming. Like PIC18FXXJ Flash devices, the actual Configuration registers are parity-protected against spurious changes.

CODE PROTECTION

All PIC18F and PIC24F devices incorporate some variety of code protection system that prevents programmed areas from being altered, or read back at any time, except during code execution.

Implementation of this feature on PIC18F devices varies from family to family in terms of how the program memory area is divided into individual blocks and the size of these blocks. Some PIC18F device families provide for separate boot block protection, for a small segment at the bottom of the program memory, in order to create a secure area for a bootloader program. All devices with data EEPROMs also provide code protection for this area as a separate block. Protection for all blocks is configured independently with three bits:

- CPn, to prevent reads and writes from outside of the device
- WRTn, to prevent all table writes
- EBTRn, to prevent table read operations executed from another code block

In contrast, PIC24F code protection is all-or-nothing: the entire program memory is either protected as one block or it is not. Protection is configured by two bits: GCP, which functions as the PIC18F CPn bit, and GWRP, which functions as the PIC18F WRTn.

ON-CHIP VOLTAGE REGULATOR

PIC24F devices use low-power circuitry for their digital logic. To ensure proper power distribution between the core and peripheral I/O systems, these devices also include an on-chip voltage regulator. The regulator allows the device to be powered from a single voltage source at the upper end of the V_{DD} range, and provides the option to power the logic core separately if needed. The regulator is hardware configured by tying an external regulator pin to V_{DD} or ground. For complete details, refer to the appropriate device data sheet.

The on-chip regulator is a departure from previous PIC18F devices which used common V_{DD} and V_{SS} busses for all on-chip circuits. The exception is the PIC18FXXJ Flash family which uses the same regulator system. Applications originally designed for these devices will be pin compatible with an equivalent PIC24F device, at least from the standpoint of power.

PIC24F PERIPHERAL SET

I/O Ports

PIC24F ports are very similar to PIC18F but have noteworthy differences. Both devices have data PORT, LAT and TRIS registers. Both analog and digital peripherals are multiplexed onto ports. Ports on the PIC24F feature more pins on more ports that have configurable open-drain and interrupt-on-change (CNx) input options.

TABLE 18: COMPARISON BETWEEN PIC18F AND PIC24F PORT FEATURES

Feature Description	PIC18F	PIC24F
Voltage Inputs Above V_{DD} + 0.3V Permitted	No	Digital-only pins
Pull-up Option	PORTB (all pins)	Multiple pins, individually selectable
Interrupt-on-Change	PORTB (3 pins)	Multiple pins, individually selectable
Control Registers	PORTx, LATx and TRISx	PORTx, LATx, TRISx and ODCx
Configured to Inputs on Reset	Yes	Yes

For both architectures, pins configured as analog inputs are read as '0' when the associated PORTx bit is read by the CPU and the output driver is disabled. This prevents shoot-through current in the port output driver.

Reading the PORTx register will return the digital representation of the pin voltage. This value will reflect an override from either a peripheral or external pin voltage.

Peripherals with digital outputs require the TRISx bit to be cleared in software. While the peripheral is enabled, the pin output can be read with the PORTx register. Enabling the peripheral may affect the output's slew rate and/or drive strength. Peripherals with the ability to tri-state the output will override the TRISx register, although the change will not affect the TRISx register value. Refer to the product data sheet for more information on peripheral outputs.

Bidirectional peripherals can override the port output but not the input. Setting the TRISx bit to '1' allows the peripheral to control the data direction. Reading the PORTx register will provide the port levels. The output may be affected by the peripheral's slew rate or drive strength overrides.

Reading the PORTx register will read the digital representation of the voltages on the pins. Reading the LATx register, reads the latched output value. Writing the PORTx register or the LATx register will write the LATx register.

On all Resets, the TRISx bits are set, causing the ports to configure as inputs and, where possible, analog inputs. This minimizes the pin load onto the application and the potential for shoot-through current in the digital port inputs.

NEW PIC24F FEATURES

- Open-drain inputs, when enabled, are active for both port and peripheral activities. The I^2C™ pins already have this function and are unaffected by the ODCx register setting.
- The Parallel Master Port (PMP) inputs are unique for the input buffer and can be software selected as either a TTL or Schmitt Trigger (ST) input buffer. This feature is available through the PADCFGx register.
- Digital-only pins are 5.5V input tolerant and are convenient for interfacing into 5V components.
- Additional pins are available for interrupt-on-change and open-drain configurations.
- The PMP is now a separate peripheral and not integrated into the I/O port functions as was the PIC18F Parallel Slave Port (PSP). Please see the product data sheet for additional information.
- Pull-ups are only available on PORTB in PIC18F devices and have a single enable bit to control all pull-ups. PIC24F supports pull-ups that can be selectively enabled on individual pins.
- Interrupt-on-change on PIC18F devices is also implemented with a single control bit for three PORTB pins. PIC24F interrupt-on-change pins can all be individually enabled.

UNSUPPORTED PIC18F FEATURES

The PIC24F port architecture does not permit the port output to drive the peripheral input. The user must either configure the pin as a peripheral input, or port output, but not both.

MIGRATION CONSIDERATIONS

- Enabling a digital or analog input or output onto a pin with a configurable open-drain option will not cause the pull-up to be automatically disabled. The pull-up is not disabled if the pin has a configurable open-drain option. It is important to disable the pull-up in software when it is not needed.
- Pins without an analog function can tolerate input voltages up to 5.5V. This can minimize hardware changes when migrating from a PIC18F device. A higher voltage output can be created by adding an external pull-up resistor on the pin and writing a zero to the data latch. Setting the TRIS bit will pull the output up to the supply voltage and clearing the TRIS bit will output a digital zero.
- Drive strength, slew rate and input voltage thresholds can change automatically when a peripheral is enabled. It is important to review the specific data sheets for differences between devices.
- Most of the input buffers for PIC24F devices are Schmitt Trigger (ST). Verify that the output levels of associated components meet the ST input voltage thresholds.

Timers (Timer1, Timer2/3 and Timer4/5)

PIC24F timers support all of the PIC18F nanoWatt features, which include Asynchronous and Synchronous Counter modes, Timer modes and 32 kHz crystal support. PIC24F timers are designed to have more generic functions, whereas PIC18F timers are intended for dedicated purposes, such as PWM clock sources or a generic timer function. Each PIC24F 16-bit timer has a dedicated period register, selectable prescaler and period match flag, and can operate in either Counter or Timer mode. PIC24F timers add new features, such as the ability to combine two timers into a single 32-bit timer and a timer gate option.

TABLE 19: COMPARISON BETWEEN PIC18F AND PIC24F TIMER FEATURES

Feature Description	PIC18F	PIC24F
Timer Width	8/16-bit	16/32-bit
General Purpose Timer Mode	All timers	All timers
Asynchronous Counter Mode	Timer1 and 3	Timer1, 2 and 4
Synchronous Counter Mode	Timer0, 1 and 3	All timers
Period Register	Timer2 and 4	All timers
32 kHz Crystal Support	Timer1	Timer1
Timer Gate Option	No	All timers
Prescaler	All timers	All timers
Postscaler	Timer2 and Timer4	No
Special Event Trigger	Yes	Yes
System Clock Source Rate	FOSC/4	FOSC/2

Although all timer functionality of the PIC18F nanoWatt devices is available in the PIC24F family, different functions may reside in different timers. Like the PIC18F, all PIC24F timers can operate using the system clock as the clock source. Where Asynchronous Counter mode was available in only Timer1 and 3 for PIC18F, this function is available in Timer1, 2 and 4 for PIC24F. Synchronous Counter mode is provided in all PIC24F timers, as well as providing a dedicated period register. The period register resets the timer on a clock cycle match and sets the interrupt flag in both architectures. However, when using the PIC24F period register, ensure the timer gate option is disabled.

Prescalers are available in both architectures. PIC18F timers have different prescale ranges, but PIC24F timers use a standard range for all timers (1:1, 1:8, 1:64 and 1:256).

Both architectures support 32 kHz crystals through Timer1, which can be optionally used as an accurate, low-power system clock source. Refer to the **"Oscillator"** and **"Power-Saving Features"** sections on pages 19 and 23, respectively, for more information on using Timer1 as a system clock.

Other device similarities include:

- Timers can operate if the device is in Sleep mode when the timer is configured for Asynchronous Counter mode or Timer1 32 kHz Crystal mode.
- All timer modes are available in Idle mode since the peripheral has a clock source.
- Both Sleep and Idle mode can be exited on a timer interrupt event.
- A Special Event Trigger is available to start an A/D conversion. Refer to the product data sheet for which timer(s) is capable of generating a Special Event Trigger.

TABLE 20: PIC18F AND PIC24F BIT EQUIVALENT TIMER FUNCTIONS

Functionality	PIC18F	PIC24F
Timer Enable	TMRxON	TON
Double Read/Write Buffer Mode	RD16	T32
Timer1 System Clock Status	T1RUN	COSC2:COSC0 = 100 (OSCCON<14:12>)
Timer Prescale Select	TxCKPSx	TCKPSx
Timer1 Oscillator Enable	T1OSCEN	SOSCEN (OSCCON<1>)
Timer External Clock Synchronization	$\overline{\text{TxSYNC}}$	TSYNC
Timer Clock Source Select	TMRxCS	TCS
Timer Postscale Select	TxOUTPSx	N/A

NEW PIC24F FEATURES

The most apparent difference between PIC18F and PIC24F timers is the size. All PIC24F timers are 16-bit and can be paired together to form a 32-bit timer. In 32-bit mode, a holding register latches the upper 16 bits when the lower 16 bits are read. The upper 16 bits are written first, which are latched until the lower half is written. This feature allows all 32 bits to be written or read without concern of a register rollover.

A timer gate option is available for PIC24F timers. This option enables the timer, when the TxCK pin is high, to set the interrupt flag on a high-to-low transition on the TxCK pin. A period match will not set the interrupt flag.

UNSUPPORTED PIC18F FEATURES

- Postscalers are not implemented in any of the PIC24F timers. Therefore, the timer interrupt flag is set on every period match.
- In PIC18F devices, Timer1 and Timer3 can both use the Timer1 oscillator, a 32 kHz crystal clock source. For PIC24F devices, only Timer1 can use the oscillator.
- The PIC24F Timer1 has only one power mode for the 32 kHz crystal oscillator. This differs from PIC18F devices which feature two oscillator gain options, set by the LPT1OSC Configuration bit.

MIGRATING A TYPICAL SETUP

In these examples, the procedure for Timer1 is shown. The process for other timers is very similar, particularly for PIC24F devices.

To configure the PIC18F Timer1 for operation:

1. Select the timer prescaler and postscaler (if available) ratio using the T1CKPS and TOUTPS bits.
2. Set the clock source using the TMR1CS bit.
3. Set or clear the $\overline{\text{T1SYNC}}$ bit to configure synchronous or asynchronous operation.
4. Load the timer period value into the PR1 register (if available).
5. If interrupts are required, set the interrupt enable bit, TXIE. Use the priority bit, TXIP, to set the interrupt priority.
6. Set the TMR1ON bit (= 1).

To configure the PIC24F Timer1 for operation:

1. Select the timer prescaler ratio using the TCKPS1:TCKPS0 bits.
2. Set the Clock and Gating modes using the TCS and TGATE bits.
3. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
4. Load the timer period value into the PR1 register.
5. If interrupts are required, set the interrupt enable bit, T1IE. Use the priority bits, T1IP2:T1IP0, to set the interrupt priority.
6. Set the TON bit (= 1).

MIGRATION CONSIDERATIONS

- PIC24F peripheral clocks derived from the system clock are twice the frequency of PIC18F devices (FOSC/2 versus FOSC/4). Be sure to account for the increased peripheral clock speed when the clock source is derived from the system clock.
- The range of prescaler settings is fixed for all PIC24F timers; PIC18F prescaler options differ from timer to timer. Both 16-bit and 32-bit timer options increase the time and resolution available to applications.
- PIC24F timers do not implement interrupt postscalers; the timer's interrupt flag will be set on every period match. If a PIC18F timer application uses a postscaler, it may be necessary to change the PIC24F timer's operation (time base or prescaler) to get the same results.
- Ensure PIC24F timer reads are performed on the entire 16-bit word. Byte reads of timer registers are not supported and will return '0's.
- PIC18F Real-Time Clock/Calendar (RTCC) applications use either Timer1 or Timer3 to maintain time information. PIC24F devices have a dedicated RTCC module which leaves Timer1 available for other tasks.

Capture/Compare/PWM (CCP and ECCP)

The PIC24F capture and compare modules exhibit the same features as the PIC18F nanoWatt CCP and ECCP peripherals, but differ in how the features are implemented. The capture module is capable of capturing timer values at the same pin edge intervals. The compare peripheral can initialize the pin at the same states and can generate a Special Event Trigger to start an A/D conversion. PWM functions are incorporated within the compare peripheral and can support multiple PWM outputs, and auto-shutdown, called Fault protection in PIC24F devices.

TABLE 21: COMPARISON BETWEEN PIC18F AND PIC24F CCP/ECCP FEATURES

Feature Description	PIC18F	PIC24F
Configurable Timer Sources	Yes	Yes
Capture Pin Prescale	1, 4, 16	Every edge, 1, 4, 16
Capture Buffer	No	Yes
Capture Timer Width	16 bits	16 bits
Selectable Captures per Interrupt	No	No
Stop in CPU Idle Mode	No	Yes
Selectable Output Compare Pin States	High, Low, Toggle	High, Low, Toggle
Special Event Trigger	Yes	Yes
Number of PWM Ouputs per Peripheral	4	1
Auto-Shutdown	Yes	Yes
Half-Bridge/Full-Bridge PWM Support	Yes	Yes
PWM Dead-Band Support	Yes	Yes

The PIC24F input capture and output compare modules can use either Timer2 or Timer3, where PIC18F can use either Timer1 or 3. Capture events can be generated on every rising, falling, 4th rising and 16th rising edge of the ICx pin.

With Single Compare Match mode selected, both architectures can select the initial state of the OCx pin. Upon the match, the pin can either transition or toggle. For each mode and in both architectures, the output compare interrupt flag is set.

All of the PIC18F PWM modes are supported by the PIC24F family. The significant difference is that each output compare peripheral can generate only one output. Therefore, half-bridge requires two peripherals and full-bridge requires four. Each output compare module can be configured for active-high or active-low output using the OCM2:OCM0 bits (OCxCON<2:0>).

PIC24F PWM mode is an extension of the output compare peripheral. This mode is similar to the Single Output Compare mode and with the addition of Fault protection pins, OCFA and OCFB, can stop the pulse train. Similar to the PIC18F PWM mode, the duty cycle is selected by the OCxR register, which is reloaded with OCxRS register contents on every match. The associated output compare interrupt flag is also set. The PWM period for both architectures is controlled by the period register, PRx, associated with the selected timer.

The PIC18F auto-shutdown option is supported by the PIC24F Fault protection option. Instead of using an external interrupt pin, INTx, the OCFA or OCFB pin is used. A low pulse on the Fault input pin will cause the pulse train to automatically stop and the Output Compare Fault Flag, OCxIF, to set. After the Fault pin returns to a digital high voltage and the Output Compare mode is selected again in the OCxCON register, the PWM pulse train is automatically restarted.

PIC18F ECCP features an option to insert a premature dead-band time in Half-Bridge mode to avoid shoot-though current in the switches, which turn off slower than they turn on. This delay is controlled by the PIC18F PDC6:PDC0 bits. This feature is not supported by the PIC24F output compare module since only one output is associated with each peripheral. To implement this feature in a PIC24F device, two timers are assigned to two compare peripherals. The difference in the initial timer's values is the desired dead-band time. The duty cycle and period are loaded for each compare module, generating a half-bridge output with dead band similar to the PIC18F definition. An advantage of the PIC24F solution is that the dead-band time is not limited to a 64-bit increment of the instruction clock, TCY.

In Full-Bridge mode, the PIC18F PWM can insert a delay between the high and low side switch activation to prevent shoot-through. This delay is selected by the Timer2 postscaler setting. The PIC24F solution is similar to the solution described in the half-bridge scenario, except that two output compare peripherals are assigned to each timer instead of one. An advantage to the PIC24F solution is that the delay is not limited to Timer2 postscaler settings.

TABLE 22: ECCP BIT NAMING CONVENTION AND FUNCTIONALITY

Functionality	PIC18F	PIC24F
Input Capture Timer Select	T3CCPx	ICTMR
Output Compare Timer Select	T3CCPx	OCTSEL
PWM Fault Condition Status	ECCPxASE	OCFLT
ECCP Mode Select bits	CCPxM<3:0> (CCPxCON<3:0>)	ICM<2:0> (Input Capture) OCM<2:0> (Output Compare)
PWM Fault Condition Enable	ECCPxAS<2:0>	OCM<2:0>

NEW PIC24F FEATURES

- **Independent Modules and Logic:** The PIC24F input capture and output compare peripherals are independent from each other and can be individually configured. This is in contrast to the PIC18F module which required configuration as either a capture, compare or PWM device at any time.
- **Optional Capture FIFO Buffer:** This allows the storage of up to four input capture timer values. When enabled, buffer empty and overflow status bits are available for buffer management.
- **Input Capture Options:** Input capture can be configured to generate events on all pin transitions (both rising and falling, not just either/or).
- **Capture Interrupt Prescaler:** The input capture module can be configured to generate an interrupt on every capture event, as well as every 2nd, 3rd and 4th event.
- **Dual Compare Mode:** PIC24F output compare peripherals feature two compare registers, OCxR and OCxRS. The additional register in Dual Compare mode can generate two pin transitions within the period to generate a wider variety of output signals.

UNSUPPORTED PIC18F FEATURES

The PIC24F input compare peripheral always generates an output on the ICx pin which is selectable in software. For PIC18F, an Interrupt Only mode is available which permits the CCPx pin to remain a port pin. The ICx pin can not be an I/O with the input capture peripheral enabled for PIC24F devices.

Hardware only option to clear auto-shutdown condition is not supported in PIC24F devices. An auto-shutdown condition is cleared automatically in hardware when the auto-shutdown condition terminates, but PIC24F devices also require the OCxCON register to be reinitialized.

A single bit, PxM1, can reverse the full-bridge PWM output direction. Since the PIC24F PWM outputs are controlled by independent output compare peripherals, switching the direction requires a different approach. By swapping the timer assignments for each output compare peripheral, using the OCTSEL bits, the output will change direction.

PIC18F Timer2 and 4 postscale settings select how many period matches will occur before the interrupt flag is set. This option is available when these timers are used for PWM clock sources. Since PIC24F timers do not have a postscaler, a counter can be used to accumulate timer interrupt events.

PWM Fault protection is only available on the OCFA and OCFB pins. A comparator output can not be used for this function. During the Fault condition, the output compare pins are placed in a high-impedance state. An option does not exist to select the output state in software, such as the PIC18F architecture.

MIGRATING A TYPICAL SETUP

Configuring the PIC18F ECCP module for PWM operation typically requires these steps:

1. Configure the PWM pins, PxA and PxB (and PxC and PxD, if used), as inputs by setting the corresponding TRIS bits.
2. Set the PWM period by loading the PRx register.
3. If auto-shutdown is required, do the following:
   a) Disable auto-shutdown (ECCPxASE = 0).
   b) Configure source (FLTx, Comparator 1 or Comparator 2).
   c) Wait for non-shutdown condition.
4. Configure the ECCP module for the desired PWM mode and configuration by loading the CCPxCON register with the appropriate values:
   a) Select one of the available output configurations and direction with the PxM1:PxM0 bits.
   b) Select the polarities of the PWM output signals with the CCPxM3:CCPxM0 bits.
5. Set the PWM duty cycle by loading the CCPRxL register and CCPxCON<5:4> bits.
6. For Half-Bridge Output mode, set the dead-band delay by loading ECCPxDEL<6:0> with the appropriate value.
7. If auto-shutdown operation is required, load the ECCPxAS register:
   a) Select the auto-shutdown sources using the ECCPxAS2:ECCPxAS0 bits.
   b) Select the shutdown states of the PWM output pins using the PSSxAC1:PSSxAC0 and PSSxBD1:PSSxBD0 bits.
   c) Set the ECCPxASE bit (ECCPxAS<7>).
   d) Configure the comparators using the CMCON register.
   e) Configure the comparator inputs as analog inputs.
8. If auto-restart operation is required, set the PxRSEN bit (ECCPxDEL<7>).
9. Configure and start TMRx:
   a) Clear the TMRx interrupt flag bit by clearing the TMRxIF bit.
   b) Set the TMRx prescale value by loading the TxCKPS bits (TxCON<1:0>).
   c) Enable Timerx by setting the TMRxON bit (TxCON<2>).
10. Enable PWM outputs after a new PWM cycle has started:
   a) Wait until TMRx overflows (TMRxIF bit is set).
   b) Enable the ECCP/PxA, PxB, PxC and/or PxD pin outputs by clearing the respective TRIS bits.
   c) Clear the ECCPxASE bit (ECCPxAS<7>).

The equivalent procedure for the PIC24F output compare module is as follows:

1. Set the PWM period by writing to the selected Timer Period register (PRx).
2. Set the PWM duty cycle by writing to the OCxRS register.
3. Write the OCxR register with the initial duty cycle.
4. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
5. Configure the output compare module for one of two PWM operation modes by writing to the Output Compare mode bits OCM<2:0> (OCxCON<2:0>).
6. Set the TMRx prescale value and enable the time base by setting TON (TxCON<15>) = 1.

MIGRATION CONSIDERATIONS

The PIC24F output compare peripheral clock source is based on FOSC/2 which differs from the PIC18F FOSC/4 clock source. Ensure the equations available in the product data sheet are used for the various calculations.

If the PIC24F capture time base increments every instruction cycle, the captured count value will be the value that was present 1 or 2 instruction cycles past the time of the event on the ICx pin. This time delay is a function of the actual ICx edge event, related to the instruction cycle clock and delay, associated with the input capture logic. If the input clock to the capture time base is prescaled, then the delay in the captured value can be eliminated.

Note:	The maximum PWM resolution is TOSC for PIC18F devices and TCY for PIC24F devices.

Serial Peripheral Interface (SPI)

The PIC24F SPI peripheral is a superset of the PIC18F architecture and many of the features use the same pin, register and bit names. Most of the 64-pin and larger PIC24F pin count devices have two SPI peripherals, as do the larger PIC18F pin count devices. The PIC24F SPI peripheral is considered a “stand-alone” peripheral, where PIC18F devices incorporate the SPI function into the larger Master Synchronous Serial Port (MSSP) peripheral, which also includes I^2C.

The functionality of a synchronous USART is supported in the PIC24F SPI, leaving the asynchronous, LIN and IrDA® functions for the PIC24F UART. See the **“Universal Asynchronous Receiver/Transmitter (UART)”** section on page 38 for additional PIC18F EUSART and PIC24F UART differences.

Both SPI peripherals support Master and Slave mode, selectable active clock edge and clock Idle state. Master mode can select when data is sampled at a variety of clock rates. The slave select feature is also common between the two architectures.

A comparison of SPI module features is provided in Table 23.

BIT NAME CHANGES AND MAPPING

Many of the PIC24F bit and register names will look familiar to a PIC18F user, as shown in Table 24.

TABLE 23: COMPARISON BETWEEN PIC18F AND PIC24F SPI FEATURES

Feature Description	PIC18F	PIC24F
Selectable Clock Polarity	Yes	Yes
Selectable Clock Edge	Yes	Yes
2-Wire Support	Yes	Yes
Slave Select Option	Yes	Yes
FIFO Buffer	No	Yes
Transfer Data Width	8	8/16
Frame Mode Support	No	Yes
Synchronous USART Functionality in SPI	No	Yes

TABLE 24: SPI BIT NAMING CONVENTION AND FUNCTIONALITY

Functionality	PIC18F	PIC24F
Clock Polarity	CKE	CKE
Clock Edge	CKP	CKP
SPI Data Buffer	SSPxBUF	SPIxBUF
Transfer or Reception States	BF	SPITBF and SPIRBF
SPI Event Occurs	SSPxIF Interrupt Flag	SPIxIF
Overflow Status bit	SSPOV	SPIROV
Sampling Time	SMP	SMP
SPI Modes and Frequency	SSPM<3:0>	MSTEN PPRE<1:0> SPRE<2:0> SSEN
SDO Disable	TRIS<SDO>	DISSDO

NEW PIC24F FEATURES

The following new PIC24F SPI features provide additional flexibility and reduced CPU overhead.

- **SPI FIFO (First-In, First-Out) Buffer:** Probably the most significant enhancement, the buffer is capable of queuing up multiple transfers without additional CPU intervention. It can be configured to behave as a single or multi-level buffer, and can track the number of pending transactions and the where the last incoming data came from (CPU or SPI receiver).
- **8 and 16-Bit Transfer Support:** Both 8 and 16-bit transfers are selected by setting MODE16 (SSPxCON1<10>).
- **SPI Frame mode:** Frame mode is available by setting FRMEN (SPIxCON2<15>).
- **Optional Frame Synchronization Pulse:** An optional frame synchronization pulse is also provided with configurable polarity and edge coincidence.
- **SPI Transfers Clocked Externally:** Similar to disabling the SDO output in Master mode, SCK output can be disabled with the DISSCK bit, allowing SPI transfers to be clocked externally.
- **SPI Interrupt Flag:** SPIxIF indicates when a reception has occurred if the SPIROV bit is already set.

UNSUPPORTED PIC18F FEATURES

All SPI PIC18F features are supported in PIC24F.

MIGRATING A TYPICAL SETUP

PIC18F SPI code block migration to PIC24F does not require much effort. The following is a typical Master mode setup for PIC18F:

1. Set the TRIS bit for the SDI pin.
2. Clear the TRIS bits for the SCK and SDO pins.
3. Select the appropriate clock edge and polarity with the CKE and CKP bits
4. Select a sampling time and clock frequency
5. If interrupts are needed, clear the SSPxIF bit and set the SSPxIE and GIE bits
6. Clear the WCOL and SSPOV bits and enable the module with the SSPEN bit

Note:	Slave mode is similar, except the TRIS bit for the SCK pin is cleared, and a slave select option is available.

A comparable setup for PIC24F SPI Master mode follows a similar setup:

1. If interrupts are desired:
   a) clear the SPIxIF bit;
   b) set the SPIxIE bit;
   c) load SPIxIP with 7.
2. Select clock frequency, polarity and active edge.
3. Input sample phase and set the MSTEN bit.
4. If SDO needs to be disabled, set the DISSDO bit.
5. Clear the SPIROV bit and set the SPIEN bit (SPIxSTAT<15>) to enable the peripheral.

Note:	Slave mode is similar, except the SPIxBUF register and the SMP and MSTEN bits are cleared. Slave select option is enabled by setting the SSEN bit.

MIGRATION CONSIDERATIONS

The only significant issue when migrating is the Master mode clock frequency calculation. Because the PIC24F instruction clock is based on FOSC/2, the peripheral clock is at a different rate than the PIC18F architecture. Use the equations in the SPI chapter of the specific device data sheet to calculate the correct SPI clock speed.

Inter-Intergrated Circuit™ (I^2C™)

The PIC24F I^2C peripheral has not changed significantly from the PIC18F definition. Many of the PIC18F control and status bits have corresponding PIC24F bits. Both architectures support 7 and 10-Bit Addressing modes, general call addressing, clock stretching, 100 and 400 kHz data rates and multi-master networking.

New features available for migrating applications include independent master and slave logic, configurable address masking and Bus Repeater mode. Whether the PIC18F I^2C software implementation uses an interrupt-based state machine or status bit polling, the transition to PIC24F can be accomplished with minimal effort.

A comparison of module features is provided in Table 25.

BIT NAME CHANGES AND MAPPING

Many of the PIC18F I^2C bits can be mapped to their PIC24F equivalent. Refer to Tables 26, 27 and 28 for bit functions common to both Master and Slave modes, and Master and Slave mode specific bit functions.

TABLE 25: COMPARISON BETWEEN PIC18F AND PIC24F I^2C™ FEATURES

Feature Description	PIC18F	PIC24F
Supported Bus Speeds	100 kHz/400 kHz	100 kHz/400 kHz/1 MHz
7 and 10-Bit Addressing Mode Support	Yes	Yes
Simplified 10-Bit Addressing Mode	No	Yes
Multi-Master Support	Yes	Yes
Configurable Address Masking	Yes (up to 6 bits)	Yes (full 7 or 10 bits)
General Call Support	Yes	Yes
Reserved Address Support	Select devices only(1)	Yes
Clock Stretching Option	Yes	Yes
Bus Repeater (IPMI) Mode	No	Yes
Slew Rate Control	Yes	Yes
Disable in Idle Mode Option	No	Yes
I^2C/SMBus Input Levels	Yes	Yes

Note 1: Refer to the specific device data sheet for information on which addresses are reserved for particular devices.

TABLE 26: PIC18F AND PIC24F BIT EQUIVALENT I^2C™ FUNCTIONALITY

Functionality	PIC18F	PIC24F
Module Enable	SSPEN	I2CEN
Stop bit Detected	P	P
Start bit Detected	S	S
Disable Slew Rate Control	SMP	DISSLW
SMBus Input Levels	CKE	SMEN
Write Attempted while Transmit Register Busy	WCOL	IWCOL
Byte Received while Byte in Receive Register	SSPOV	I2COV
Receive Buffer Full	BF	RBF
Transmit Buffer Full	BF	TBF

TABLE 27: PIC18F AND PIC24F BIT EQUIVALENT I^2C™ SLAVE FUNCTIONS

Functionality	PIC18F	PIC24F
SCL Stretch/Release	CKP	SCLREL
I2CxADD 10-Bit Slave Enable	SSPM3:SSPM0 = 1111	A10M
General Call Enable	GCEN	GCEN
Data/Address bit	D/$\overline{A}$	D/$\overline{A}$
Read/Write bit	R/$\overline{W}$	R/$\overline{W}$

TABLE 28: PIC18F AND PIC24F BIT EQUIVALENT I²C™ MASTER FUNCTIONS

Functionality	PIC18F	PIC24F
Acknowledge Data bit	ACKDT	ACKDT
Acknowledge Sequence Enable	ACKEN	ACKEN
Receive Enable	RCEN	RCEN
Stop Condition Enable	PEN	PEN
Repeated Start Condition Enable	RSEN	RSEN
Start Condition Enable	SEN	SEN
Acknowledge Status	ACKSTAT	ACKSTAT
Transmit Status	TRMT	TRSTAT
Bus Collision	BCLIF	BCL and MI2CxIF

NEW PIC24F FEATURES

The following new PIC24F I²C features provide additional flexibility and reduced CPU overhead.

- **Independent Master/Slave Logic:** In the PIC24F implementation, I²C master and slave functions have independent logic and functions. Each of these sections can operate independently and simultaneously, and generate its own interrupts.
- **Bus Repeater Mode**: The PIC24F module can function as a “passive” repeater, capable of responding to all addresses and performing automatic clock stretching. In this mode, reserved I²C addresses are also Acknowledged.
- **General Call Support:** A General Call Status bit, GCSTAT (I2CxSTAT<9>), eliminates the need to read SPIxBUF to confirm the address.
- **Full Slave Address Masking:** The slave logic in the PIC24F module allows masking of all address bits in both 7-Bit and 10-Bit Addressing modes. The PIC18F module only permits masking on bits 0 through 7 in both modes.
- **Address Detection:** The 10-Bit Addressing mode has been simplified for the Update Address bit, UA. Update address is no longer necessary and address matches are automatically detected with the available ADD10 (I2CxSTAT<8>) status bit.

UNSUPPORTED PIC18F FEATURES

The PIC24F I²C peripheral does not have a Firmware Controlled Master mode configuration (similar to the PIC18F mode, where SSPM<3:0> = 1011). As a result, PIC24F I²C firmware implementation must use port input (V_{IH} and V_{IL}) levels.

MIGRATING A TYPICAL SETUP

PIC18F I²C implementations vary greatly between various applications. Due to this high degree of variance, an adequate summary of each potential implementation would be complex and still not cover every application. A good approach to migrating I²C code is to translate each PIC18F I²C event into its PIC24F equivalent, using Tables 26, 27 and 28. Also, review the new PIC24F features which can significantly reduce code size and simplify the software solution.

MIGRATION CONSIDERATIONS

Differences between the PIC24F and PIC18F I²C peripherals can lead to some complications during migration. A common issue is to not account for the address shift in the I2CxMSK and I2CxADD registers. The Least Significant I²C address bit in the PIC18F SSPxADD register is bit ‘1’, where in the PIC24F I2CxADD register, it is bit ‘0’. A simple shift prior to loading or after reading the address will resolve the difference.

In Sleep or Idle mode, some PIC24F devices cannot wake-up on a Slave mode address match and may not update the D/$\overline{A}$ bit in Slave Transmit mode. Refer to the appropriate PIC24F data sheet to determine whether these features are supported.

Defining Master mode baud rates has changed for PIC24F devices. Since the PIC24F uses an instruction clock based on F_{OSC}/2, the peripheral clock rate differs by a factor of two. Refer to the specific device data sheet for the baud rate equations and tables.

Universal Asynchronous Receiver/Transmitter (UART)

The new PIC24F Universal Asynchronous Receiver Transmitter (UART) module is a full-duplex, asynchronous system with enhanced functionality over the PIC18F Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module. Some of the new PIC24F UART module features include increased baud rate, parity hardware support (even/odd/none) for 8-bit data, selectable Idle polarity, software Stop in Idle, improved interrupt options and hardware flow control supports. Table 29 summarizes the comparison between the PIC18F EUSART and PIC24F UART.

Note:	The PIC24F UART does not support synchronous communications. If synchronous serial communication is required, use the SPI module instead.

TABLE 29: COMPARISON BETWEEN PIC18F AND PIC24F UART FEATURES

Features	PIC18F EUSART	PIC24F UART
Asynchronous (full-duplex) Operation with: - Auto-Wake-up on Character Reception - Auto-Baud Calibration - 12-Bit Break Character Transmission	Yes	Yes
Full-Duplex 8 or 9-Bit Data Transmission through TX and RX pins	Yes	Yes
Support for 9-Bit Mode with Address Detect (9th bit = 1)	Yes	Yes
Hardware Flow Control Option with CTS and RTS pins	No	Yes
Number of Stop bits	1	1 or 2
Selectable Idle Polarity	No	Yes
Baud Rate Generator	Dedicated 8-bit/16-bit	16-bit
BRG Prescaler	Yes	Yes
Baud Rate range	300 bps to 115 kbps at 0.25-10 MIPS	15 bps to 1 Mbps at 16 MIPS
IrDA® Encoder and Decoder Logic	No	Yes
16x Baud Clock Output for IrDA Support	No	Yes
FIFO Transmit Data Buffer	No	Yes
FIFO Receive Data Buffer	No	Yes
Loopback Mode for Diagnostic Support	No	Yes
Hardware Parity Support (8-bit data)	No (can be implemented in software)	Yes
Parity Error Detection	No	Yes
Hardware Sync Byte Generation	Yes	Yes
Support for Sync and Break Characters	Yes	Yes
Wake-up Enable	Yes	Yes
Framing and Buffer Overrun Error Detection	Yes	Yes
Software Stop in Idle Option	No	Yes
Interrupt Options (see **"Interrupt Controller"** on page 17)	Transmit and Receive	Transmit, Receive and UART Error Event

TABLE 30: UART BIT NAMING CONVENTION AND FUNCTIONALITY

Functionality	PIC18F	PIC24F
Auto Baud Detect Enable bit/Auto-Baud Enable bit	ABDEN	ABAUD
Address Detect Enable bit/Address Character Detect bit (bit 8 of received data = 1)	ADDEN	ADDEN
High Baud Rate Select bit: (PIC18F high/low-speed, PIC24F 4x/16x baud clock)	BRGH	BRGH
Continuous Receive Enable bit	CREN (receiver only)	UARTEN (transmitter and receiver)
Framing Error bit/Framing Error Status bit (read-only)	FERR	FERR
Overrun Error bit/Receive Buffer Overrun Error Status bit (clear/read-only)	OERR	OERR
Receive Operation Idle Status bit/Receiver Idle bit (read-only)	RCIDL	RIDLE
EUSART Receive Interrupt Flag bit (PIC18F) UART1 Receiver Interrupt Flag Status bit (PIC24F)	RCIF	UxRXIF
Receive Buffer Data Available bit (read-only)		URXDA
9th bit of Received Data bit	RX9D	UxRXREG<8> (word read needed)
9-Bit Receive Enable bit/Parity and Data Selection bits	RX9	PDSEL1:PDSEL0
9-Bit Transmit Enable bit/Parity and Data Selection bits	TX9	
Send Break Character bit	SENDB	UTXBRK
Transmit Shift Register Status bit (PIC18F) Transmit Shift Register Empty bit (read-only, PIC24F)	TRMT	TRMT
Transmit Enable bit	TXEN	UTXEN
EUSART Transmit Interrupt Flag bit/UARTx Transmitter Interrupt Flag Status bit	TXIF	UxTXIF
9th bit of Transmit Data	TX9D	UxTXREG<8> (word write needed)
Wake-up Enable bit/Wake-up on Start bit Detect during Sleep Mode Enable bit	WUE	WAKE

UNSUPPORTED PIC18F FEATURES

The following are PIC18F EUSART features that are not supported by the newer PIC24F UART module:

- FOSC/[64 (n + 1)] baud rate.
- Synchronous Master and Slave Communication modes are not supported in the PIC24F UART. To use Synchronous mode in PIC24F devices, use the SPI module.
- The PIC18F EUSART option of changing between low speed and high speed by changing the BRGH bit between '0' and '1' is not supported in PIC24F. The target will need to change the baud rate for PIC24F.

TRANSMISSION SETUP (8-BIT DATA MODE)

To set up the PIC18F EUSART for a transmission in 8-Bit Data mode:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Select Asynchronous or Synchronous mode via the SYNC bit and setting bit, SPEN.
3. If interrupts are desired, set enable bit TXxIE.
4. If 9-bit transmission is desired, set transmit bit, TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit, TXEN, which will also set bit, TXxIF.
6. If 9-bit transmission is selected, the 9th bit should be loaded in bit, TX9D.
7. Load data to the TXREGx register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

The equivalent steps for the PIC24F UART are:

1. Set up the UART:
   a) Write appropriate values for data, parity and Stop bits.
   b) Write appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt).
4. Write data byte to lower byte of UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with next rising edge of the baud clock.
5. Alternately, the data byte may be transferred while UTXEN = 0, and then the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated through the interrupt control bit, UTXISELx.

RECEPTION SETUP (8-BIT DATA MODE)

To set up the PIC18F EUSART for a reception in 8-bit data mode:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Select Asynchronous or Synchronous mode via the SYNC bit and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RCxIE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RCxIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCxIE, was set.
7. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREGx register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

The equivalent steps for the PIC24F UART are:

1. Setup the UART
   a) Write appropriate values for data, parity and Stop bits.
   b) Write appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UART.
3. A receive interrupt will be generated when one or more data characters has been received as per interrupt control bit, URXISELx.
4. Read the OERR bit to determine if an overrun error has occurred.

Note:	The OERR bit must be reset in software.

5. Read UxRXREG.

MIGRATION CONSIDERATIONS

When migrating from a PIC18F design to a PIC24F, the following must be taking into consideration:

- Because the fundamental instruction cycle rate is different (FOSC/2 for PIC24F, FOSC/4 for PIC18F), projects that are being ported from PIC18F to PIC24F will need to have baud rates recalculated.
- For PIC18F, the maximum USART baud rate is FOSC/4, and the minimum is FOSC/(64 * 65536). For PIC24F, the maximum UART baud rate possible (BRGH = 1) is FCY/4 (for BRG = 0) and the minimum baud rate possible is FCY/(4 * 65536).
- Routines for 9-bit communications will need to be modified. PIC18F USARTs require the 9th bit to be read from, or written to another register, besides RXREGx or TXREGx. PIC24F UARTs can perform 9-bit operations with a single write to UxTXREG or a read from UxRXREG.

10-Bit A/D Converter

The PIC24F A/D has significant improvements in performance and features over the PIC18F implementation (see Table 31). Improvements include a conversion rate of 500 ksps (samples per second), Automatic Channel Scan mode, 16-bit conversion result buffer, selectable Buffer Fill modes, four result format options and individually selectable A/D inputs. These features improve A/D performance and flexibility, and minimize CPU overhead.

A comparison of available features is presented in Table 31.

TABLE 31: COMPARISON BETWEEN PIC18F AND PIC24F A/D FEATURES

Feature Description	PIC18F	PIC24F
Resolution	10-bit	10-bit
Conversion Throughput (ksps)	100	500
Available Voltage Reference Sources	Internal and External	Internal and External
DONE Status bit	Yes	Yes
Selectable A/D Clock Divider	Yes	Yes
A/D RC Oscillator	Yes	Yes
Auto-Sample	Yes	Yes
Programmable Sample Time	Yes	Yes
Individually Selectable Analog Inputs	No	Yes
Special Event Trigger	Yes	Yes
Multiple Channel Scan	No	Yes
FIFO Buffer	No	Yes
Multiple Result Formats	No	Yes
Differential Channel (comparative) Conversion	No	Yes

BIT NAME CHANGES AND MAPPING

The PIC24F and PIC18F A/D module have similar features. Both have a 10-bit, Successive Approximation Register (SAR) A/D capable of using a combination of reference pins (V_{REF+} and V_{REF-}) and analog power pins (AV_{DD} and AVss) for the reference voltages. Both product lines feature an A/D conversion status bit, selectable A/D clock divider, dedicated A/D RC, auto-sampling with configurable sample time, analog/digital input selection and run-time selectable A/D input. Conversions can be initiated by software, an external interrupt or an output compare event.

The associated control and status bits for the common features may have different names but perform similar functions, as shown in Table 32.

TABLE 32: CONTROL AND STATUS BIT NAMING CONVENTION AND FUNCTIONALITY

Functionality	PIC18F Name	PIC24F Name
Bits select a combination of AV_{DD} and V_{REF+} for the positive voltage reference and AV_{SS} and V_{REF-} for the negative voltage reference.	VCFG2:VCFG0	VCFG2:VCFG0
Indicates when the conversion is complete, an associated interrupt is also available.	GO/$\overline{\text{DONE}}$	DONE
One of two A/D clock sources can be selected, either the dedicated A/D RC oscillator or the divider of the system clock. An associated interrupt is also available.	ADCS2:ADCS0	ADRC, ADCS7:ADCS0
Auto-sampling time defined.	ACQT2:ACQT0	SAMC4:SAMC0
To configure an I/O for an analog input, disable the digital input buffers.	PCFG3:PCFG0	PCFG15:PCFG0
Specifies the pin at which the analog conversion will be performed.	CHS3:CHS0	CSSL15:CSSL0
A conversion can be initiated from an external interrupt or output compare event.	CCPxM3:CCPxM0 (CCPxM:<3:0>)	SSRC2:SSRC0

NEW PIC24F A/D FEATURES

Many of the improvements of the PIC24F A/D are likely to add value to applications migrating from the PIC18F architecture.

- **Enhanced Sampling Speed and Throughput:** The minimum A/D clock period has been reduced from 750 ns to 75 ns, providing conversion rates up to 500 ksps (thousand samples per second).
- **Flexible Results Buffering:** The results buffer can be configured as two 8-word buffers or one 16-word buffer. In Two 8-Word Buffer mode, results can be written either sequentially through each buffer or alternately between the two buffers.
- **Output Formats:** Conversion data can be provided in one of four user-selectable data formats.
- **Input Scanning**: Multiple analog channels can be successively scanned across two sets of multiplexed inputs using the Automatic Channel Scan mode option.
- **Individual Pin Configuration:** Each of the analog input pins can be individually selected for Analog Input mode, instead of as part of a sequence of pins.
- **Differential Conversion:** This conversion mode allows the direct measurement of the voltage difference between two analog inputs.

UNSUPPORTED PIC18F FEATURES

All PIC18F A/D converter features are supported in the PIC24F A/D.

MIGRATING A TYPICAL SETUP

Preparing for a PIC24F A/D conversion is very similar to setting up a PIC18F conversion. One or more pins are configured as an analog input and the A/D parameters select the acquisition time and conversion clock. The A/D is enabled. If an interrupt is needed, the A/D interrupt flag is cleared and the interrupt is enabled. After the minimum required acquisition time has expired, either in software or automatically by the module, the A/D conversion is started. When the interrupt occurs or the DONE bit is read high, the result is available in the designated result register.

To prepare the PIC24F A/D for a comparable PIC18F conversion, perform the following:

1. Select which pins will become A/D inputs (AD1PCFG<15:0>).
2. Determine A/D voltage references (AD1CON2<15:13>).
3. Select A/D clock divider (AD1CON3<7:0>) or dedicated RC (AD1CON3<15>).
4. Choose whether sampling will begin automatically after the previous conversion or by software (AD1CON1<2>).
5. Select the trigger source to begin the conversion, either by software, an INT0 transition, Timer3 compare or automatically by the module (AD1CON1<7:5>).
6. Determine whether the results will be stored as integer, signed integer, fractional or signed fractional format (AD1CON1<9:8>).
7. To match the PIC18F architecture, generate an interrupt on every conversion by clearing the SMPI (AD1CON2<5:2>) bits.
8. Enable the module (AD1CON1<15>).
9. If desired, enable the A/D interrupt by clearing the interrupt flag (IFS0<13>), setting the interrupt enable bit (IEC0<13>) and setting the interrupt priority to 7 (IPC3<6:4>).

MIGRATION CONSIDERATIONS

- The analog input voltage range of the PIC24F is smaller than PIC18F devices due to the smaller V_{DD} range of the PIC24F device family. The voltage range of the PIC24F is 2.0V to 3.6V, whereas the range for most PIC18F devices is 2.0V to 5.5V.

Note:	For both architectures, the analog voltage range must not exceed 0.3V above V_{DD} or below V_{SS}.

- The analog input impedance for the PIC24F module is 2.5 kΩ. Many PIC18F devices are 10 kΩ, although 2.5 kΩ is recommended.
- For the PIC24F module, the module's internal sampling capacitor is 4.4 pF, typical; for the PIC18F module, it is 25 pF, typical. The reduced capacitance increases the affect of the external capacitance on the analog input.
- When configuring the A/D to use the conversion clock, several factors will affect the PIC24F divider selection. These include the reduced T_{AD} and the instruction rate. The PIC24F conversion clock is based on the instruction clock, $T_{CY}/2$, where the PIC18F is based on $F_{OSC}/2$. Due to different instruction rates for a given system clock frequency, the smallest period for the PIC24F A/D clock divider is one F_{OSC} period and two for PIC18F.

Comparator and Comparator Voltage Reference Modules

The comparator modules on the PIC18F and PIC24F devices share many of the same features. Each has two comparators with various configuration selections. The PIC18F version has only eight selections, one of which is disabling comparators. The PIC24F version is more configurable, allowing individual control over many of the options that are fixed on PIC18F. In addition to the comparator module, both PIC18F and PIC24F offer a comparator voltage reference based on a resistor ladder circuit.

TABLE 33: COMPARISON BETWEEN PIC18F AND PIC24F COMPARATOR MODULE FEATURES

Feature Description	PIC18F	PIC24F
Comparators	2	2
Output Inversion Control	Yes	Yes
Comparator Output on I/O pin	Available in 3 modes	Always available, individually enabled
Separate Comparator Enables	Available in 1 mode	Always available, individually enabled
Multiple Input Selections	Available in 1 mode	Always available, individually configurable for all inputs
Detecting Individual Comparator Output Changed States	Tracked in firmware by user	Tracked in hardware

TABLE 34: COMPARISON BETWEEN PIC18F AND PIC24F COMPARATOR VOLTAGE REFERENCE MODULE FEATURES

Feature Description	PIC18F	PIC24F
Resistor Ladder	16-tap	16-tap
Two Selectable Ranges	Yes	Yes
Selectable Reference from Analog Power or MCU Power	Yes	Yes
Voltage Reference Output Enable	Yes	Yes

TABLE 35: COMPARATOR AND COMPARATOR VOLTAGE BIT NAMING CONVENTION AND FUNCTIONALITY

Functionality	PIC18F	PIC24F
Comparator Output	CxOUT	CxOUT
Comparator Output Inversion	CxINV	CxINV
Comparator Input Switch	CIS	No
Comparator Mode	CMx	No
Stop in Idle Mode	No	CMIDL
Comparator Event	No	CxEVT
Comparator Enable	No	CxEN
Comparator Output Enable	No	CxOUTEN
Comparator Negative Input Configure	No	CxNEG
Comparator Positive Input Configure	No	CxPOS

NEW PIC24F COMPARATOR MODULE FEATURES

- **Increased Flexibility in Configuration:** The PIC24F comparator module allows individual control over the inverting and non-inverting inputs to each of the comparators, comparator enables, comparator output inversion and output on I/O pins. In contrast, the comparator module in PIC18F devices was limited to eight fixed configurations, one of which has both comparators disabled and another has the comparators in Reset. The PIC24F module can reproduce all but one of these configurations.
- **Individual Comparator Status Tracking:** The PIC24F module has two individual comparator event control bits that can be used for detecting when an individual comparator output changes states. The PIC18F devices require firmware to maintain information about the status of the output bits to determine which comparator has changed states.

NEW PIC24F COMPARATOR VOLTAGE REFERENCE MODULE FEATURES

The comparator voltage reference module is identical to that of the PIC18F module. The control registers and the bits within them are identically named and have the same functions.

UNSUPPORTED PIC18F FEATURES

The PIC18F module has two configurations, where the V_{IN+} input on both comparators are tied together and share one I/O pin. To achieve one of these two modes on PIC24F, the V_{IN+} input on each comparator must be tied externally together using one more I/O pins than the comparable PIC18F configuration.

TYPICAL SETUP SEQUENCE

In this example, the PIC18F comparator module will be configured as two independent comparators with outputs enabled and Comparator 1 output inverted. The comparator voltage reference module is configured for output enabled (outputs onto V_{IN+} pin of Comparator 1) and set for 0.25 * V_{DD}. All referenced bits are in the CMCON and CVRCON registers.

1. Set the Comparator mode bits, CM2:CM0 to '011', which enables two independent comparators with output enabled.
2. Clear the C2INV bit for a non-inverted output for Comparator 2.
3. Set the C1INV bit to invert Comparator 1 output.
4. Enable the comparator voltage reference module and its output by setting the CVREN and CVROE bits.
5. Select the CV_{RSRC}/32 step size for the voltage range.
6. Select AV_{DD} and AV_{SS} as the supply voltage by clearing the CVRSS bit.
7. Select 0.25 * V_{DD} as the voltage reference value by clearing bits, CVR3:CVR0.

This sequence configures the PIC24F comparator and comparator voltage reference modules to operate the same as the previous PIC18F setup. Control bits are in the CMCON (for comparator) and CVRCON (for voltage reference) registers.

1. Enable the comparators by setting the C1EN and C2EN bits.
2. Enable the comparator outputs by setting the C1OUTEN and C2OUTEN bits.
3. Clear the C2INV bit for a non-inverting output for Comparator 2.
4. Set the C1INV bit to invert Comparator 1 output.
5. Connect Comparator 1 negative input pin to V_{IN-} by clearing C1NEG.
6. Connect Comparator 1 positive input pin to the comparator voltage reference by clearing C1POS.
7. Connect Comparator 2 negative input to V_{IN-} by clearing C2NEG.
8. Connect Comparator 2 positive input is to V_{IN+} by setting C2POS.
9. Enable the comparator voltage reference module and its output by setting the CVREN and CVROE bits.
10. Select the CV_{RSRC}/32 step size for the voltage range.
11. Select AV_{DD} and AV_{SS} as the supply voltage by clearing the CVRSS bit.
12. Select 0.25 * V_{DD} as the voltage reference value by clearing bits, CVR3:CVR0.

SUMMARY

The preceding represents the most important considerations for moving a legacy, 8-bit PIC18F application to a PIC24F platform. Since there are so many ways for implementing a given solution, it is impractical in this space to list every possible code or algorithm translation into its 16-bit equivalent. Instead, this document should be thought of as a checklist of the most important differences between those features shared by both device families.

Nor does this document consider all the possibilities of the PIC24F architecture and its expanded peripheral set. Features, such as the Programmable CRC Generator, hardware Real-Time Clock/Calendar and Parallel Master Port, may offer the chance for making an existing application smaller and more efficient. Ultimately, the user will need to review their application's design and implementation, and make the decisions as to how their code and hardware design must be changed to work on the new platform.

REFERENCES

For more specific information on the device families used as the basis for this document, please refer to the following data sheets:

- *"PIC18F8722 Family Data Sheet"* (DS39646)
- *"PIC24FJ128GA Family Data Sheet"* (DS39747)

Detailed information on oscillator design and trouble-shooting for PICmicro microcontroller applications is provided in these Microchip application notes:

- *AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices"* (DS00826)
- *AN849, "Basic PICmicro® Oscillator Design"* (DS00849)
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"* (DS00943)
- *AN949, "Making Your Oscillator Work"* (DS00949)

Microchip also has additional data sheet and reference material available for PIC18F, PIC18FXXJ Flash and PIC24F devices that may be helpful in planning the migration of an application. For a complete listing of available materials, users are encouraged to visit the Microchip corporate web site at:

www.microchip.com

NOTES:

APPENDIX A: MAPPING PIC18F TO PIC24F INSTRUCTIONS

This section describes how PIC18F assembly instructions map to PIC24F assembly instructions. Keep in mind that the addressing structure of the two product lines has changed and this will affect how many of these instructions are mapped. See the **"Addressing Modes"** section on page 5 of this document for more details on the different addressing modes. The mapping provided does not cover all possible permutations of the different addressing modes. Please be aware that if indirect addressing is available for an instruction, a single example is provided in the mapping chart. Also note that this chart only maps PIC18F instructions to PIC24F instructions. It does not highlight many of the new capabilities of the PIC24F product line.

Below is a description of the differences in notation of the instruction sets between the two product lines.

Key:

`Wn`, `Ws`, `Wb` and `Wd`: any of the working registers. For PIC24F, WREG refers exclusively to W0.

`[Ws]`: use Ws as a pointer; similar to using INDF0 in PIC18F.

`[Ws++]`: use Ws as a pointer and increment it after accessing the variable it points to. Similar to using POSTINC0 in PIC18F.

`[Ws + Wd]`: use Ws as a pointer and Wd as an offset to the pointer. Similar to using PLUSW0 in PIC18F.

`.B` or `.D`: immediately following an instruction mnemonic indicates that the instruction will operate on a byte or double word (4 bytes), rather than a word (2 bytes).

`Varn`: the address of a variable *n* in data memory space.

`#litn`: a literal value of length *n*.

`Expr`: an address in program memory space.

TABLE A-1: PIC18F ASSEMBLY INSTRUCTIONS AND THEIR PIC24F EQUIVALENTS

PIC18F Instruction	PIC18F Example(s)	PIC24F Equivalent Instruction(s)	Notes
ADDFSR	ADDFSR 0, #lit6	ADD #lit10, Wn	1) This is part of the extended instruction set and may not be available on all PIC18F devices. 2) Most W registers can be used as indirect pointers (see **"Addressing Modes"** section on page 5).
ADDLW	ADDLW #lit8	ADD #lit10, Wn	
ADDULNK(1)	ADDULNK#lit6	LNK #lit14	
ADDWF	ADDWF var1, w	ADD var1, WREG	This instruction can only operate on W0.
	ADDWF var1, f	ADD var1	This instruction can only operate on W0.
	ADDWF INDF0, w	ADD Wb, [Ws], Wb	
	ADDWF INDF0, f	ADD Wb, [Ws], [Ws]	
ADDWFC	ADDWF var1, w	ADDC var1, WREG	This instruction can only operate on W0.
	ADDWF var1, f	ADDC var1	This instruction can only operate on W0.
	ADDWF INDF0, w	ADDC Wb, [Ws], Wb	
	ADDWF INDF0, f	ADDC Wb, [Ws], [Ws]	
ANDLW	ANDLW #lit8	AND #lit10, Wn	Wn = Wn OR #lit10.
		AND Wb, #lit5, Wd	Wd = Wb OR #lit5.
ANDWF	ANDWF var1, w	AND var1, WREG	This instruction can only operate on W0.
	ANDWF var1, f	AND var1	This instruction can only operate on W0.
	ANDWF INDF0, w	AND Wb, [Ws], Wb	
	ANDWF INDF0, f	AND Wb, [Ws], [Ws]	
BC	BC Expr	BRA C, Expr	
BCF	BCF var1, #lit3	BCLR var1, #lit4	
		BCLR Ws, #lit4	
	BCF INDF0, #lit3	BCLR [Ws], #lit4	

Note 1: This instruction is part of the PIC18F extended instruction set. It is not available in earlier PIC18F devices or PIC18F devices operating in Legacy mode (XINST Configuration bit is unprogrammed).

TABLE A-1: PIC18F ASSEMBLY INSTRUCTIONS AND THEIR PIC24F EQUIVALENTS (CONTINUED)

PIC18F Instruction	PIC18F Example(s)	PIC24F Equivalent Instruction(s)	Notes
BN	BN Expr	BRA N, Expr	
BNC	BNC Expr	BRA NC, Expr	
BNN	BNN Expr	BRA NN, Expr	
BNOV	BNOV Expr	BRA NOV, Expr	
BNZ	BNZ Expr	BRA NZ, Expr	
BOV	BOV Expr	BRA OV, Expr	
BRA	BRA Expr	BRA Expr	
BSF	BSF var1, #lit3	BSET var1, #lit4	
		BSET Ws, #lit4	
	BSF INDF0, #lit3	BSET [Ws], #lit4	
BTFSC	BTFSC var1, #lit3	BTSC var1, #lit4	
		BTSC Ws, #lit4	
	BTFSC INDF0, #lit3	BTSC [Ws], #lit4	
BTFSS	BTFSS var1, #lit3	BTSS var1, #lit4	
		BTSS Ws, #lit4	
	BTFSS INDF0, #lit3	BTSS [Ws], #lit4	
BTG	BTG var1, #lit3	BTG var1, #lit4	
		BTG Ws, #lit4	
	BTG INDF0, #lit3	BTG [Ws], #lit4	
BZ	BZ Expr	BRA Z, Expr	
CALL	CALL Expr	CALL Expr	
CALLW(1)	CALLW	CALLW Wn	
CLRF	CLRF var1	CLR var1	This instruction can only operate on W0.
		CLR WREG	
		CLR Ws	
	CLRF INDF0	CLR [Ws]	
CLRWDT	CLRWDT	CLRWDT	
COMF	COMF var1, w	COM var1, WREG	This instruction can only operate on W0.
	COMF var1, f	COM var1	
	COMF INDF0, w	COM [Ws], Wb	
	COMF INDF0, f	COM [Ws], [Ws]	
CPFSEQ	CPFSEQ var1	(No exact equivalent)	
	CPFSEQ INDF0	CPSEQ Wb, [Wn]	Compares Wb to the contents pointed to by [Wn].
CPFSGT	CPFSGT var1	(No exact equivalent)	
	CPFSGT INDF0	CPSGT Wb, [Wn]	Compares Wb to the contents pointed to by [Wn].
CPFSLT	CPFSLT var1	(No exact equivalent)	
	CPFSLT INDF0	CPSLT Wb, [Wn]	Compares Wb to the contents pointed to by [Wn].
DAW	DAW	DAW.B Wn	Wn is the destination of the operation.

Note 1: This instruction is part of the PIC18F extended instruction set. It is not available in earlier PIC18F devices or PIC18F devices operating in Legacy mode (XINST Configuration bit is unprogrammed).

TABLE A-1: PIC18F ASSEMBLY INSTRUCTIONS AND THEIR PIC24F EQUIVALENTS (CONTINUED)

PIC18F Instruction	PIC18F Example(s)	PIC24F Equivalent Instruction(s)	Notes
DCFSNZ	DCFSNZ var1, w	(No exact equivalent)	
	DCFSNZ var1, f		
	DCFSNZ INDF0, w		
	DCFSNZ INDF0, f		
DECF	DECF var1, w	DEC var1, WREG	This instruction can only operate on W0.
	DECF var1, f	DEC var1	
	DECF INDF0, w	DEC [Ws], Wd	
	DECF INDF0, f	DEC [Ws], [Ws]	
DECFSZ	DECFSZ var1, w	(No exact equivalent)	
	DECFSZ var1, f		
	DECFSZ INDF0, w		
	DECFSZ INDF0, f		
GOTO	GOTO Expr	GOTO Expr	
INCF	INCF var1, w	INC var1, WREG	This instruction can only operate on W0.
	INCF var1, f	INC var1	
	INCF INDF0, w	INC [Ws], Wd	
	INCF INDF0, f	INC [Ws], [Ws]	
INCFSZ	INCFSZ var1, w	(No exact equivalent)	
	INCFSZ var1, f		
	INCFSZ INDF0, w		
	INCFSZ INDF0, f		
INFSNZ	INFSNZ var1, w	(No exact equivalent)	
	INFSNZ var1, f		
	INFSNZ INDF0, w		
	INFSNZ INDF0, f		
IORLW	IORLW #lit8	IOR #lit10, Wn	Wn = Wn OR #lit10.
		IOR Wb, #lit5, Wd	Wd = Wb OR #lit5.
IORWF	IORWF var1, w	IOR var1, WREG	This instruction can only operate on W0.
	IORWF var1, f	IOR var1	
	IORWF INDF0, w	IOR Wb, [Ws], Wb	
	IORWF INDF0, f	IOR Wb, [Ws], [Ws]	
LFSR	LFSR 0, var1	MOV #var1, Wd	Most W registers can be used as indirect pointers (see **"Addressing Modes"** section on page 5).
MOVF	MOVF var1, w	MOV var1, Wn	
	MOVF var1, f	MOV var1	
	MOVF INDF0, w	MOV [Ws], Wd	
		MOV.D [Ws], Wd	
	MOVF INDF0, f	MOV [Ws,], [Ws]	
MOVFF	MOVFF var1, var2	(No exact equivalent)	
	MOVFF INDF0, var1		
	MOVFF INDF0, INDF1	MOV [Ws], [Wd]	
MOVLB	MOVLB var1	N/A	PIC24F does not require banking.

Note 1: This instruction is part of the PIC18F extended instruction set. It is not available in earlier PIC18F devices or PIC18F devices operating in Legacy mode (XINST Configuration bit is unprogrammed).

TABLE A-1: PIC18F ASSEMBLY INSTRUCTIONS AND THEIR PIC24F EQUIVALENTS (CONTINUED)

PIC18F Instruction	PIC18F Example(s)	PIC24F Equivalent Instruction(s)	Notes
MOVLW	MOVLW #lit8	MOV #lit16, Wd	
MOVSF(1)	MOVSF [#lit7], var1	(No exact equivalent)	
	MOVSF [#lit7], INDF0		
MOVSS(1)	MOVSS [#lit7], [#lit7]	(No exact equivalent)	
MOVWF	MOVWF var1	MOV Ws, var1	
	MOVWF INDF0	MOV Ws, [Wd]	
MULLW	MULLW #lit8	MUL.UU Wb, #lit5, Wn	1) The result of this operation is stored in Wn:Wn + 1. 2) Note that the PIC24F equivalent is a #lit5 instead of a #lit8 like the PIC18F instruction.
MULWF	MULWF var1	MUL var1	1) This instruction can only operate on W0. 2) The result of this operation is stored in W2:W3.
	MULWF INDF0	MUL.UU Wb, [Ws], Wn	The result of this operation is stored in Wn:Wn + 1.
NEGF	NEGF var1	NEG var1	
	NEGF INDF0	NEG [Ws], [Ws]	
NOP	NOP	NOP	
		NOPR	
POP	POP	POP var1	Note that all POP operations on PIC24F require a destination address and thus, the result of the POP can be saved.
		POP Wn	
		POP [Wn]	
		POP.D Wn	
		DEC2 W15	If the results of the POP do not need to be maintained.
PUSH	PUSH	PUSH var1	Note that all PUSH operations on PIC24F require a source address and thus, values other then the PC can be pushed.
		PUSH Wn	
		PUSH [Wn]	
		PUSH.D Wn	
PUSHL(1)	PUSHL #lit8	(No exact equivalent)	
RCALL	RCALL Expr	RCALL Expr	
RESET	RESET	RESET	
RETFIE	RETFIE	RETFIE	
	RETFIE FAST		
RETLW	RETLW #lit8	RETLW #lit10, Wn	Wn is the destination of the operation.
RETURN	RETURN	RETURN	
RLCF	RLCF var1, w	RLC var1, WREG	
	RLCF var1, f	RLC var1	
	RLCF INDF0, w	RLC [Ws], Wd	
	RLCF INDF0, f	RLC [Ws], [Ws]	

Note 1: This instruction is part of the PIC18F extended instruction set. It is not available in earlier PIC18F devices or PIC18F devices operating in Legacy mode (XINST Configuration bit is unprogrammed).

TABLE A-1: PIC18F ASSEMBLY INSTRUCTIONS AND THEIR PIC24F EQUIVALENTS (CONTINUED)

PIC18F Instruction	PIC18F Example(s)	PIC24F Equivalent Instruction(s)	Notes
RLNCF	RLNCF var1, w	RLNC var1, WREG	
	RLNCF var1, f	RLNC var1	
	RLNCF INDF0, w	RLNC [Ws], Wd	
	RLNCF INDF0, f	RLNC [Ws], [Ws]	
RRCF	RRCF var1, w	RRC var1, WREG	
	RRCF var1, f	RRC var1	
	RRCF INDF0, w	RRC [Ws], Wd	
	RRCF INDF0, f	RRC [Ws], [Ws]	
RRNCF	RRNCF var1, w	RRNC var1, WREG	
	RRNCF var1, f	RRNC var1	
	RRNCF INDF0, w	RRNC [Ws], Wd	
	RRNCF INDF0, f	RRNC [Ws], [Ws]	
SETF	SETF var1	SETM var1	
		SETM WREG	This instruction can only operate on W0.
		SETM Ws	
	SETF INDF0	SETM [Ws]	
SLEEP	SLEEP	PWRSAV #lit1	If #lit1 = 0, then it is Sleep mode. If #lit1 = 1, then the part goes into Idle mode.
SUBFSR(1)	SUBFSR 0, #lit6	SUB #lit10, Wn	Most W registers can be used as indirect pointers (see **"Addressing Modes"** section on page 5).
SUBFWB	SUBFWB var1, w	SUBBR var1, WREG	This instruction can only operate on W0.
	SUBFWB var1, f	SUBBR var1	This instruction can only operate on W0.
	SUBFWB INDF0, w	SUBB Wd, [Ws], Wd,	
	SUBFWB INDF0, f	SUBB Wd, [Ws], [Ws]	
SUBLW	SUBLW #lit8	SUB #lit10, Wd	
SUBULNK(1)	SUBULNK#lit6	(No exact equivalent)	Note that the PIC24F instruction, ULNK, is very similar but not equivalent.
SUBWF	SUBWF var1, w	SUB var1, WREG	This instruction can only operate on W0.
	SUBWF var1, f	SUB var1	This instruction can only operate on W0.
	SUBWF INDF0, w	SUBR Wd, [Ws], Wd,	
	SUBWF INDF0, f	SUBR Wd, [Ws], [Ws]	
SUBWFB	SUBWFB var1, w	SUBB var1, WREG	This instruction can only operate on W0.
	SUBWFB var1, f	SUBB var1	This instruction can only operate on W0.
	SUBWFB INDF0, w	SUBBR Wd, [Ws], Wd	
	SUBWFB INDF0, f	SUBBR Wd, [Ws], [Ws]	
SWAPF	SWAPF WREG	SWAP Wn	
	SWAPF var1, w	(No exact equivalent)	
	SWAPF var1, f		
	SWAPF INDF0, w		
	SWAPF INDF0, f		

Note 1: This instruction is part of the PIC18F extended instruction set. It is not available in earlier PIC18F devices or PIC18F devices operating in Legacy mode (XINST Configuration bit is unprogrammed).

TABLE A-1: PIC18F ASSEMBLY INSTRUCTIONS AND THEIR PIC24F EQUIVALENTS (CONTINUED)

PIC18F Instruction	PIC18F Example(s)	PIC24F Equivalent Instruction(s)	Notes
TBLRD*	TBLRD*	TBLRDL [Ws], Wd	1) [Ws] is a pointer to the program memory address. 2) Wd is the destination (or pointer to the destination) of the read.
		TBLRDL [Ws], [Wd]	
TBLRD*-	TBLRD*-	TBLRDL [Ws--], Wd	1) [Ws] is a pointer to the program memory address. 2) Wd is the destination (or pointer to the destination) of the read.
		TBLRDL [Ws--], [Wd]	
TBLRD*+	TBLRD*+	TBLRDL [Ws++], Wd	1) [Ws] is a pointer to the program memory address. 2) Wd is the destination (or pointer to the destination) of the read.
		TBLRDL [Ws++], [Wd]	
TBLRD+*	TBLRD+*	TBLRDL [++Ws], Wd	1) [Ws] is a pointer to the program memory address. 2) Wd is the destination (or pointer to the destination) of the read.
		TBLRDL [++Ws], [Wd]	
TBLWT*	TBLWT*	TBLWTL Ws, [Wd]	1) Ws is the source (or pointer to the source) of data that is written. 2) Wd is the destination (or pointer to the destination) of the write.
		TBLWTL [Ws], [Wd]	
TBLWT*-	TBLWT*-	TBLWTL Ws, [Wd--]	1) Ws is the source (or pointer to the source) of data that is written. 2) Wd is the destination (or pointer to the destination) of the write.
		TBLWTL [Ws], [Wd--]	
TBLWT*+	TBLWT*+	TBLWTL Ws, [Wd++]	1) Ws is the source (or pointer to the source) of data that is written. 2) Wd is the destination (or pointer to the destination) of the write.
		TBLWTL [Ws], [Wd++]	
TBLWT+*	TBLWT+*	TBLWTL Ws, [++Wd]	1) Ws is the source (or pointer to the source) of data that is written. 2) Wd is the destination (or pointer to the destination) of the write.
		TBLWTL [Ws], [++Wd]	
TSTFSZ	TSTFSZ var1	(No exact equivalent)	
XORLW	XORLW #lit8	XOR #lit10, Wd	Wd is the destination of the operation.
		XOR Wb, #lit5, Wd	
XORWF	XORWF var1, w	XOR var1, WREG	This instruction can only operate on W0.
	XORWF var1, f	XOR var1	This instruction can only operate on W0.
	XORWF INDF0, w	XOR Wd, [Ws], Wd	
	XORWF INDF0, f	XOR Wd, [Ws], [Ws]	

Note 1: This instruction is part of the PIC18F extended instruction set. It is not available in earlier PIC18F devices or PIC18F devices operating in Legacy mode (XINST Configuration bit is unprogrammed).

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/16/06