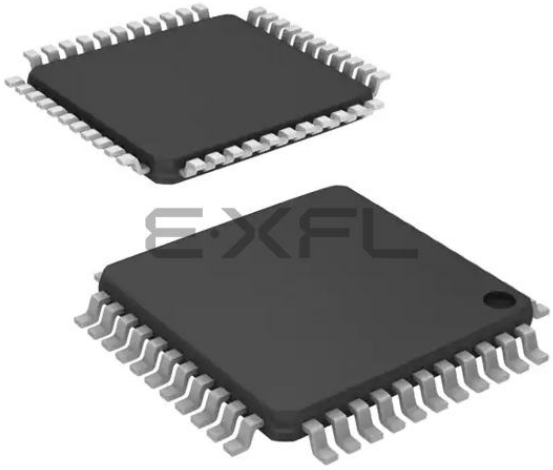


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 13x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f46j13t-i-pt

TABLE 1-3: PIC18F2XJ13 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	28-SPDIP/ SSOP/ SOIC	28-QFN			
$\overline{MCLR}$	1(2)	26(2)	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OSC1/CLKI/RA7	9	6			Oscillator crystal or external clock input.
OSC1			I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; CMOS otherwise. Main oscillator input connection.
CLKI			I	CMOS	External clock source input; always associated with pin function, OSC1 (see related OSC1/CLKI pins).
RA7(1)			I/O	TTL/DIG	Digital I/O.
OSC2/CLKO/RA6	10	7			Oscillator crystal or clock output.
OSC2			O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO			O	DIG	Main oscillator feedback output connection. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6(1)			I/O	TTL/DIG	Digital I/O.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD}); DIG = Digital output; I^2C = Open-Drain, I^2C specific

Note 1: RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

2: 5.5V tolerant.

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	MCLR Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
SPBRGH2	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
BAUDCON2	PIC18F2XJ13	PIC18F4XJ13	0100 0-00	0100 0-00	uuuu u-uu
TMR3H	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F2XJ13	PIC18F4XJ13	0000 0000	uuuu uuuu	uuuu uuuu
TMR4	PIC18F2XJ13	PIC18F4XJ13	0000 0000	uuuu uuuu	uuuu uuuu
PR4	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
T4CON	PIC18F2XJ13	PIC18F4XJ13	-000 0000	-000 0000	-uuu uuuu
SSP2BUF	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
SSP2MSK	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
SSP2STAT	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F2XJ13	PIC18F4XJ13	---- -111	---- -111	---- -uuu
PMADDRH(5)	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDOUT1H(5)	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMADDRL	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDIN1H	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDIN1L	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
TXADDRL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
TXADDRH	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- uuuu
RXADDRL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
RXADDRH	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- uuuu
DMABCL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
DMABCH	PIC18F2XJ13	PIC18F4XJ13	---- --00	---- --00	---- --uu
PMCONH	PIC18F2XJ13	PIC18F4XJ13	0--0 0000	0--0 0000	u--u uuuu
PMCONL	—	PIC18F4XJ13	000- 0000	000- 0000	uuu- uuuu
PMMODEH	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMMODEL	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDOUT2H	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 5-1 for the Reset value for a specific condition.

5: Not implemented on PIC18F2XJ13 devices.

6: Not implemented on "LF" devices.

6.1.4.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 1L. When STVREN is set, a full or underflow condition sets the appropriate STKFUL or STKUNF bit and then causes a device Reset. When STVREN is cleared, a full or underflow condition sets the appropriate STKFUL or STKUNF bit, but does not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a POR.

6.1.5 FAST REGISTER STACK (FRS)

A Fast Register Stack (FRS) is provided for the STATUS, WREG and BSR registers to provide a "fast return" option for interrupts. This stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources push values into the Stack registers. The values in the registers are then loaded back into the working registers if the `RETFIE, FAST` instruction is used to return from the interrupt.

If both low-priority and high-priority interrupts are enabled, the Stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the FRS for returns from interrupt. If no interrupts are used, the FRS can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a `CALL label, FAST` instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A `RETURN, FAST` instruction is then executed to restore these registers from the FRS.

Example 6-1 provides a source code example that uses the FRS during a subroutine call and return.

EXAMPLE 6-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST    ;STATUS, WREG, BSR
                   ;SAVED IN FAST REGISTER
                   ;STACK
        .
        .

SUB1    .
        .
    RETURN FAST    ;RESTORE VALUES SAVED
                   ;IN FAST REGISTER STACK
```

6.1.6 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures or look-up tables in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed `GOTO`
- Table Reads

6.1.6.1 Computed GOTO

A computed `GOTO` is accomplished by adding an offset to the PC. An example is shown in Example 6-2.

A look-up table can be formed with an `ADDWF PCL` instruction and a group of `RETLW nn` instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the `ADDWF PCL` instruction. The next executed instruction will be one of the `RETLW nn` instructions that returns the value, '`nn`', to the calling function.

The offset value (in WREG) specifies the number of bytes that the PC should advance and should be multiples of 2 (LSb = `0`).

In this method, only one byte may be stored in each instruction location, but room on the return address stack is required.

EXAMPLE 6-2: COMPUTED GOTO USING AN OFFSET VALUE

```
        MOVF    OFFSET, W
        CALL    TABLE
ORG     nn00h
TABLE   ADDWF   PCL
        RETLW   nnh
        RETLW   nnh
        RETLW   nnh
        .
        .
        .
```

6.1.6.2 Table Reads

A better method of storing data in program memory allows two bytes to be stored in each instruction location.

Look-up table data may be stored two bytes per program word while programming. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from the program memory. Data is transferred from program memory, one byte at a time.

Table read operation is discussed further in **Section 7.1 "Table Reads and Table Writes"**.

6.4 Data Addressing Modes

Note: The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 6.6 "Data Memory and the Extended Instruction Set"** for more information.

While the program memory can be addressed in only one way through the PC, information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in more detail in **Section 6.6.1 "Indexed Addressing with Literal Offset"**.

6.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all; they either perform an operation that globally affects the device, or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DAW.

Other instructions work in a similar way, but require an additional explicit argument in the opcode. This is known as Literal Addressing mode, because they require some literal value as an argument. Examples include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

6.4.2 DIRECT ADDRESSING

Direct Addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit literal address as their LSB. This address specifies either a register address in one of the banks of data RAM (**Section 6.3.3 "General Purpose Register File"**) or a location in the Access Bank (**Section 6.3.2 "Access Bank"**) as the data source for the instruction.

The Access RAM bit, 'a', determines how the address is interpreted. When 'a' is '1', the contents of the BSR (**Section 6.3.1 "Bank Select Register"**) are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit, 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction; their destination is either the target register being operated on or the W register.

6.4.3 INDIRECT ADDRESSING

Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as SFRs, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code using loops, such as the example of clearing an entire RAM bank in Example 6-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

EXAMPLE 6-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
        LFSR    FSR0, 0x100 ;
NEXT    CLRF    POSTINC0    ; Clear INDF
                            ; register then
                            ; inc pointer
        BTFSS   FSR0H, 1    ; All done with
                            ; Bank1?
        BRA     NEXT        ; NO, clear next
CONTINUE                    ; YES, continue
```

7.5 Writing to Flash Program Memory

The programming block is 32 words or 64 bytes. Programming one word or 2 bytes at a time is also supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 64 holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation (if WPROG = 0). All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

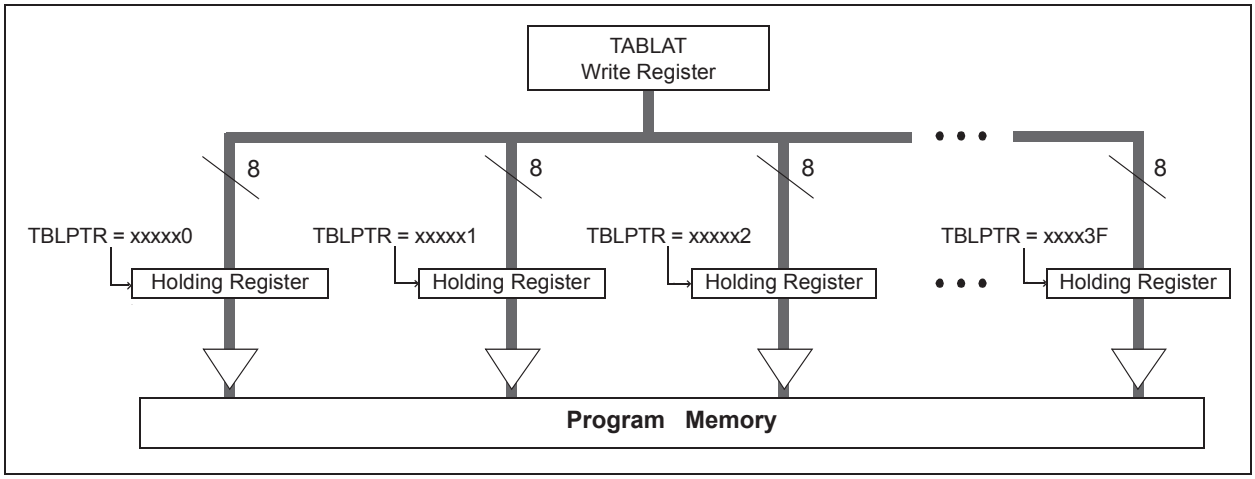
The long write is necessary for programming the internal Flash. Instruction execution is Halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

Note 1: Unlike previous PIC® devices, devices of the PIC18F47J13 Family do not reset the holding registers after a write occurs. The holding registers must be cleared or overwritten before a programming sequence.

2: To maintain the endurance of the program memory cells, each Flash byte should not be programmed more than once between erase operations. Before attempting to modify the contents of the target cell a second time, an erase of the target page, or a bulk erase of the entire memory, must be performed.

FIGURE 7-5: TABLE WRITES TO FLASH PROGRAM MEMORY



7.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 1024 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load the Table Pointer register with address being erased.
4. Execute the erase procedure.
5. Load the Table Pointer register with the address of the first byte being written, minus 1.
6. Write the 64 bytes into the holding registers with auto-increment.
7. Set the WREN bit (EECON1<2>) to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit; this will begin the write cycle.
12. The CPU will stall for the duration of the write for TIW (see parameter D133A).
13. Re-enable interrupts.
14. Repeat Steps 6 through 13 until all 1024 bytes are written to program memory.
15. Verify the memory (table read).

An example of the required code is provided in Example 7-3 on the following page.

Note: Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 64 bytes in the holding register.

9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2, PIE3). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

REGISTER 9-9: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ACCESS F9Dh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIE(1)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

bit 7 **PMPIE:** Parallel Master Port Read/Write Interrupt Enable bit(1)
1 = Enables the PMP read/write interrupt
0 = Disables the PMP read/write interrupt

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit 5 **RC1IE:** EUSART1 Receive Interrupt Enable bit
1 = Enables the EUSART1 receive interrupt
0 = Disables the EUSART1 receive interrupt

bit 4 **TX1IE:** EUSART1 Transmit Interrupt Enable bit
1 = Enables the EUSART1 transmit interrupt
0 = Disables the EUSART1 transmit interrupt

bit 3 **SSP1IE:** Master Synchronous Serial Port 1 Interrupt Enable bit
1 = Enables the MSSP1 interrupt
0 = Disables the MSSP1 interrupt

bit 2 **CCP1IE:** ECCP1 Interrupt Enable bit
1 = Enables the ECCP1 interrupt
0 = Disables the ECCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Note 1: These bits are unimplemented on 28-pin devices.

REGISTER 9-12: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4 (ACCESS F8Eh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCP10IE	CCP9IE	CCP8IE	CCP7IE	CCP6IE	CCP5IE	CCP4IE	CCP3IE
bit 7							bit 0

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

bit 7-1 **CCP10IE:CCP4IE:** CCP<10:4> Interrupt Enable bits
1 = Enabled
0 = Disabled

bit 0 **CCP3IE:** ECCP3 Interrupt Enable bit
1 = Enabled
0 = Disabled

REGISTER 9-13: PIE5: PERIPHERAL INTERRUPT ENABLE REGISTER 5 (ACCESS F91h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CM3IE	TMR8IE	TMR6IE	TMR5IE	TMR5GIE	TMR1GIE
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **CM3IE:** Comparator 3 Receive Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **TMR8IE:** TMR8 to PR8 Match Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **TMR6IE:** TMR6 to PR6 Match Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **TMR5IE:** TMR5 Overflow Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR5GIE:** TMR5 Gate Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **TMR1GIE:** TMR1 Gate Interrupt Enable bit
1 = Enabled
0 = Disabled

9.6 INTx Pin Interrupts

External interrupts on the INT0, INT1, INT2 and INT3 pins are edge-triggered. If the corresponding INTEDGx bit in the INTCON2 register is set (= 1), the interrupt is triggered by a rising edge; if the bit is clear, the trigger is on the falling edge. When a valid edge appears on the INTx pin, the corresponding flag bit and INTxIF are set. This interrupt can be disabled by clearing the corresponding enable bit, INTxIE. Flag bit, INTxIF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1, INT2 and INT3) can wake up the processor from the Sleep and Idle modes if bit, INTxIE, was set prior to going into the power-managed modes. Deep Sleep mode can wake up from INT0, but the processor will start execution from the Power-on Reset vector rather than branch to the interrupt vector.

Interrupt priority for INT1, INT2 and INT3 is determined by the value contained in the Interrupt Priority bits, INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0; It is always a high-priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (FFh → 00h) will set flag bit, TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh → 0000h) will set TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 12.0 "Timer0 Module"** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit, RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

9.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the Fast Return Stack. If a fast return from interrupt is not used (see **Section 6.3 "Data Memory Organization"**), the user may need to save the WREG, STATUS and BSR registers on entry to the Interrupt Service Routine. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine (ISR).

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF   W_TEMP                  ; W_TEMP is in virtual bank
MOVFF   STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF   BSR, BSR_TEMP           ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF   BSR_TEMP, BSR           ; Restore BSR
MOVF    W_TEMP, W               ; Restore WREG
MOVFF   STATUS_TEMP, STATUS     ; Restore STATUS
```

TABLE 10-9: PORTD I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RD0/PMD0/ SCL2	RD0	1	I	ST	PORTD<0> data input.
		0	O	DIG	LATD<0> data output.
	PMD0(1)	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	SCL2(1)	1	I	I^2C/ SMB	I^2C clock input (MSSP2 module); input type depends on the module setting.
		0	O	DIG	I^2C clock output (MSSP2 module); takes priority over port data.
RD1/PMD1/ SDA2	RD1	1	I	ST	PORTD<1> data input.
		0	O	DIG	LATD<1> data output.
	PMD1(1)	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	SDA2(1)	1	I	I^2C/ SMB	I^2C data input (MSSP2 module); input type depends on the module setting.
		0	O	DIG	I^2C data output (MSSP2 module); takes priority over port data.
RD2/PMD2/ RP19	RD2	1	I	ST	PORTD<2> data input.
		0	O	DIG	LATD<2> data output.
	PMD2(1)	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	RP19	1	I	ST	Remappable Peripheral Pin 19 input.
		0	O	DIG	Remappable Peripheral Pin 19 output.
RD3/PMD3/ RP20	RD3	1	I	ST	PORTD<3> data input.
		0	O	DIG	LATD<3> data output.
	PMD3(1)	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	RP20	1	I	ST	Remappable Peripheral Pin 20 input.
		0	O	DIG	Remappable Peripheral Pin 20 output.
RD4/PMD4/ RP21	RD4	1	I	ST	PORTD<4> data input.
		0	O	DIG	LATD<4> data output.
	PMD4(1)	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	RP21	1	I	ST	Remappable Peripheral Pin 21 input.
		0	O	DIG	Remappable Peripheral Pin 21 output.
RD5/PMD5/ RP22	RD5	1	I	ST	PORTD<5> data input.
		0	O	DIG	LATD<5> data output.
	PMD5(1)	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	RP22	1	I	ST	Remappable Peripheral Pin 22 input.
		0	O	DIG	Remappable Peripheral Pin 22 output.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; I^2C/SMB = I^2C/SMBus input buffer; x = Don't care (TRISx bit does not affect port direction or is overridden for this option).

Note 1: Only on 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).

19.4.2 FULL-BRIDGE MODE

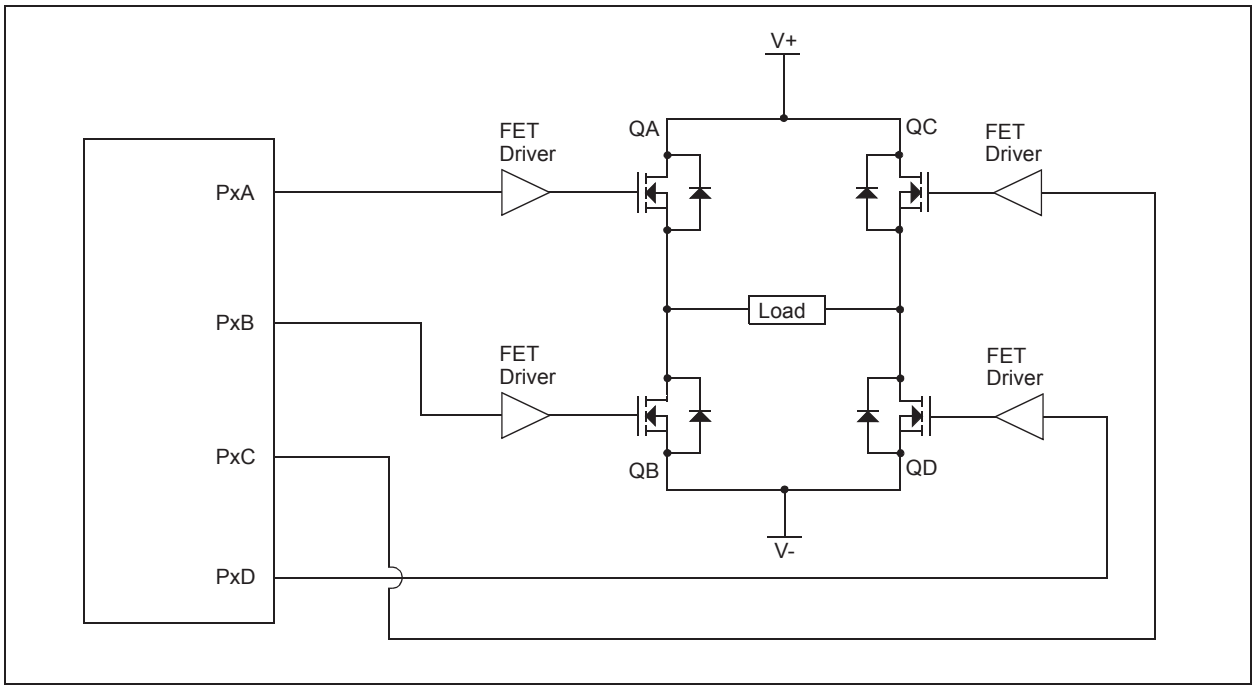
In Full-Bridge mode, all four pins are used as outputs. An example of a full-bridge application is provided in Figure 19-8.

In the Forward mode, the PxA pin is driven to its active state and the PxD pin is modulated, while the PxB and PxC pins are driven to their inactive state, as shown in Figure 19-9.

In the Reverse mode, the PxC pin is driven to its active state and the PxB pin is modulated, while the PxA and PxD pins are driven to their inactive state, as shown in Figure 19-9.

The PxA, PxB, PxC and PxD outputs are multiplexed with the port data latches. The associated TRIS bits must be cleared to configure the PxA, PxB, PxC and PxD pins as outputs.

FIGURE 19-8: EXAMPLE OF FULL-BRIDGE APPLICATION



20.5.3.4 7-Bit Address Masking Mode

Unlike 5-Bit Address Masking mode, 7-Bit Address Masking mode uses a mask of up to eight bits (in 10-bit addressing) to define a range of addresses than can be Acknowledged, using the lowest bits of the incoming address. This allows the module to Acknowledge up to 127 different addresses with 7-bit addressing, or 255 with 10-bit addressing (see Example 20-4). This mode is the default configuration of the module and is selected when MSSPMSK is unprogrammed ('1').

The address mask for 7-Bit Address Masking mode is stored in the SSPxMSK register, instead of the SSPxCON2 register. SSPxMSK is a separate, hardware register within the module, but it is not directly addressable. Instead, it shares an address in the SFR space with the SSPxADD register. To access the SSPxMSK register, it is necessary to select MSSP mode, '1001' (SSPCON1<3:0> = 1001), and then read or write to the location of SSPxADD.

To use 7-Bit Address Masking mode, it is necessary to initialize SSPxMSK with a value before selecting the I^2C Slave Addressing mode. Thus, the required sequence of events is:

1. Select SSPxMSK Access mode (SSPxCON2<3:0> = 1001).
2. Write the mask value to the appropriate SSPxADD register address (FC8h for MSSP1, F74h for MSSP2).
3. Set the appropriate I^2C Slave mode (SSPxCON2<3:0> = 0111 for 10-bit addressing, 0110 for 7-bit addressing).

Setting or clearing mask bits in SSPxMSK behaves in the opposite manner of the ADMSK bits in 5-Bit Address Masking mode. That is, clearing a bit in SSPxMSK causes the corresponding address bit to be masked; setting the bit requires a match in that position. SSPxMSK resets to all '1's upon any Reset condition, and therefore, has no effect on the standard MSSP operation until written with a mask value.

With 7-Bit Address Masking mode, the SSPxMSK<7:1> bits mask the corresponding address bits in the SSPxADD register. For any SSPxMSK bits that are active (SSPxMSK<n> = 0), the corresponding SSPxADD address bit is ignored (SSPxADD<n> = x). For the module to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

With 10-Bit Address Masking mode, SSPxMSK<7:0> bits mask the corresponding address bits in the SSPxADD register. For any SSPxMSK bits that are active (= 0), the corresponding SSPxADD address bit is ignored (SSPxADD<n> = x).

Note:	The two MSbs of the address are not affected by address masking.

EXAMPLE 20-4: ADDRESS MASKING EXAMPLES IN 7-BIT MASKING MODE

7-Bit Addressing:

SSPxADD<7:1>= 1010 000

SSPxMSK<7:1>= 1111 001

Addresses Acknowledged = ACh, A8h, A4h, A0h

10-Bit Addressing:

SSPxADD<7:0> = 1010 0000 (the two MSbs are ignored in this example since they are not affected)

SSPxMSK<5:1> = 1111 0011

Addresses Acknowledged = ACh, A8h, A4h, A0h

FIGURE 21-4: ASYNCHRONOUS TRANSMISSION

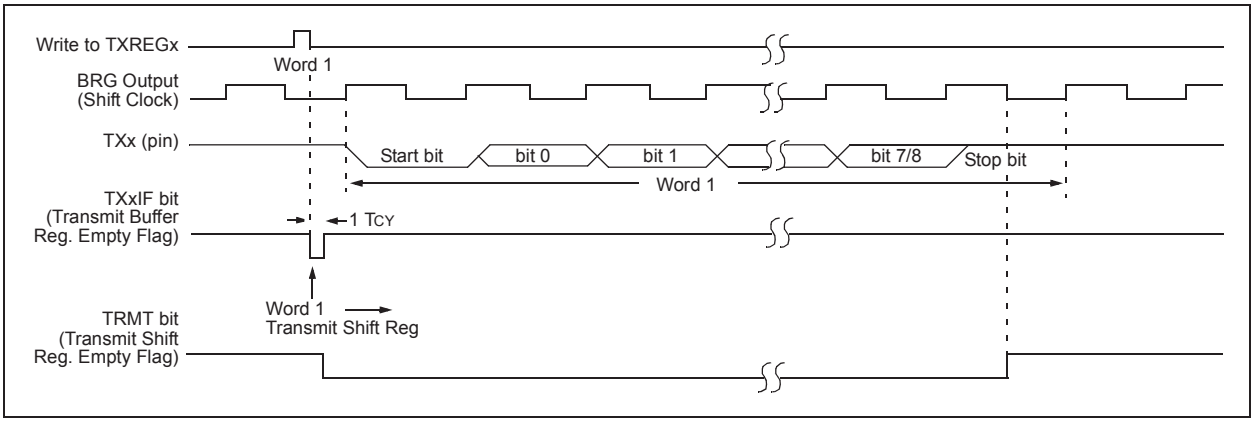
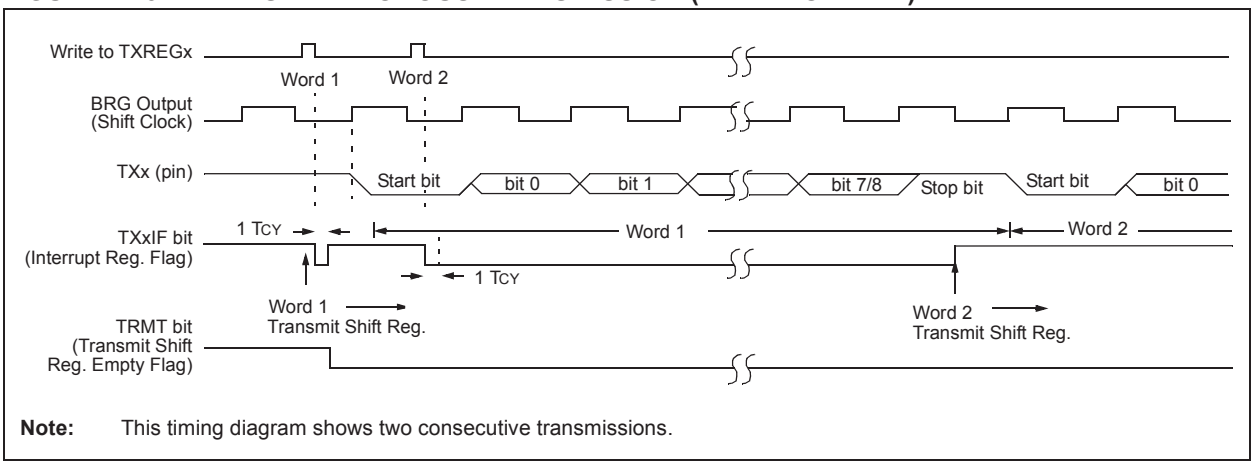


FIGURE 21-5: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 21-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PMPIF(1)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIE1	PMPIE(1)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
IPR1	PMPIP(1)	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
TXREGx	EUSARTx Transmit Register							
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte							
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte							
ODCON2	—	—	—	—	CCP10OD	CCP9OD	U2OD	U1OD

Legend: – = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: These bits are only available on 44-pin devices.

25.2 HLVD Setup

To set up the HLVD module:

1. Disable the module by clearing the HLVDEN bit (HLVDCON<4>).
2. Write the value to the HLVDL<3:0> bits that select the desired HLVD trip point.
3. Set the VDIRMAG bit to detect one of the following:
   - High voltage (VDIRMAG = 1)
   - Low voltage (VDIRMAG = 0)
4. Enable the HLVD module by setting the HLVDEN bit.
5. Clear the HLVD Interrupt Flag, HLVDIF (PIR2<2>), which may have been set from a previous interrupt.
6. If interrupts are desired, enable the HLVD interrupt by setting the HLVDIE and GIE/GIEH bits (PIE2<2> and INTCON<7>).

   An interrupt will not be generated until the IRVST bit is set.

25.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled, and will consume static current. The total current consumption, when enabled, is specified in electrical specification parameter, D022B (ΔI_{HLVD}) (**Section 30.2 "DC Characteristics: Power-Down and Supply Current PIC18F47J13 Family (Industrial)"**).

Depending on the application, the HLVD module does not need to operate constantly. To decrease the current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After doing the check, the HLVD module may be disabled.

25.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification parameter D420 (see Table 30-8 in **Section 30.0 "Electrical Characteristics"**), may be used by other internal circuitry, such as the Programmable Brown-out Reset (BOR).

If the HLVD or other circuits using the voltage reference are disabled to lower the device's current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time, T_{IRVST}, is an interval that is independent of device clock speed. It is specified in electrical specification, parameter 36 (Table 30-15).

The HLVD interrupt flag is not enabled until T_{IRVST} has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval. Refer to Figure 25-2 or Figure 25-3.

REGISTER 27-5: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
DSWDTPS3(1)	DSWDTPS2(1)	DSWDTPS1(1)	DSWDTPS0(1)	DSWDTEN(1)	DSBOREN	RTCOSC	DSWDTOSC(1)
bit 7							bit 0

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

bit 7-4 **DSWDTPS<3:0>:** Deep Sleep Watchdog Timer Postscale Select bits(1)
The DSWDT prescaler is 32. This creates an approximate base time unit of 1 ms.
1111 = 1:2,147,483,648 (25.7 days)
1110 = 1:536,870,912 (6.4 days)
1101 = 1:134,217,728 (38.5 hours)
1100 = 1:33,554,432 (9.6 hours)
1011 = 1:8,388,608 (2.4 hours)
1010 = 1:2,097,152 (36 minutes)
1001 = 1:524,288 (9 minutes)
1000 = 1:131,072 (135 seconds)
0111 = 1:32,768 (34 seconds)
0110 = 1:8,192 (8.5 seconds)
0101 = 1:2,048 (2.1 seconds)
0100 = 1:512 (528 ms)
0011 = 1:128 (132 ms)
0010 = 1:32 (33 ms)
0001 = 1:8 (8.3 ms)
0000 = 1:2 (2.1 ms)

bit 3 **DSWDTEN:** Deep Sleep Watchdog Timer Enable bit(1)
1 = DSWDT is enabled
0 = DSWDT is disabled

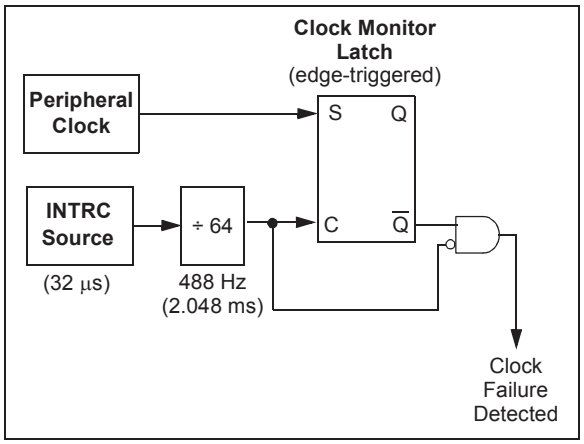
bit 2 **DSBOREN:** "F" Device Deep Sleep BOR Enable bit, "LF" Device VDD BOR Enable bit
For "F" Devices:
1 = VDD sensing BOR is enabled in Deep Sleep
0 = VDD sensing BOR circuit is always disabled
For "LF" Devices:
1 = VDD sensing BOR circuit is always enabled
0 = VDD sensing BOR circuit is always disabled

bit 1 **RTCOSC:** RTCC Reference Clock Select bit
1 = RTCC uses T1OSC/T1CKI as the reference clock
0 = RTCC uses INTRC as the reference clock

bit 0 **DSWDTOSC:** DSWDT Reference Clock Select bit(1)
1 = DSWDT uses INTRC as the reference clock
0 = DSWDT uses T1OSC/T1CKI as the reference clock

Note 1: Functions are not available on "LF" devices.

FIGURE 27-4: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while the clock monitor is still set, and a clock failure has been detected (Figure 27-5), the following results:

- The FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>);
- The device clock source is switched to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the fail-safe condition); and
- The WDT is reset.

During switchover, the postscaler frequency from the internal oscillator block may not be sufficiently stable for timing-sensitive applications. In these cases, it may be desirable to select another clock configuration and enter an alternate power-managed mode. This can be done to attempt a partial recovery or execute a controlled shutdown. See **Section 4.1.4 "Multiple Sleep Commands"** and **Section 27.4.1 "Special Considerations For Using Two-speed Start-up"** for more details.

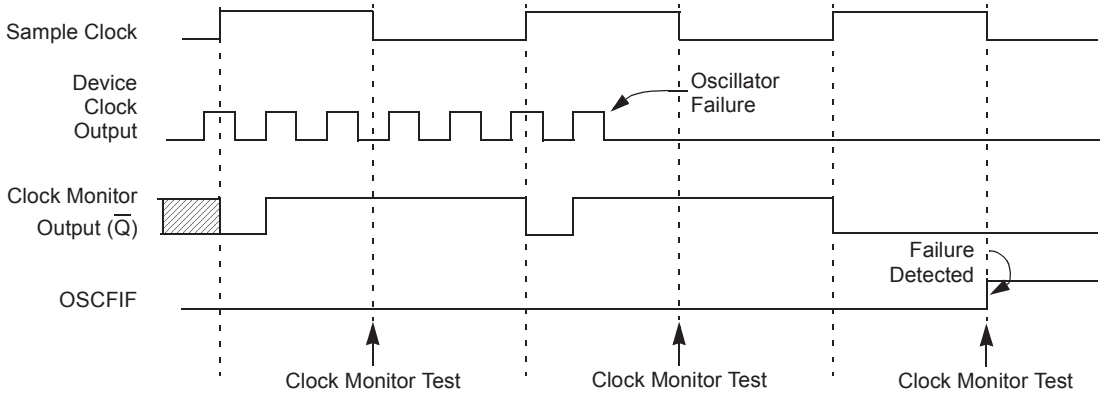
The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

27.5.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTRC clock when a clock failure is detected; this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock Monitor events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

FIGURE 27-5: FSCM TIMING DIAGRAM



Note: The device clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

BTG **Bit Toggle f**

Syntax: BTG f, b {,a}

Operands: $0 \le f \le 255$
$0 \le b < 7$
$a \in [0,1]$

Operation: $(\overline{f<b>}) \rightarrow f<b>$

Status Affected: None

Encoding:

0111	bbba	ffff	ffff

Description: Bit 'b' in data memory location 'f' is inverted.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 28.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: `BTG PORTC, 4, 0`

Before Instruction:
PORTC = 0111 0101 [75h]
After Instruction:
PORTC = 0110 0101 [65h]

BOV **Branch if Overflow**

Syntax: BOV n

Operands: $-128 \le n \le 127$

Operation: if Overflow bit is '1',
$(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0100	nnnn	nnnn

Description: If the Overflow bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example:
```
HERE    BOV   Jump
```
Before Instruction
PC = address (HERE)
After Instruction
If Overflow = 1;
PC = address (Jump)
If Overflow = 0;
PC = address (HERE + 2)

30.2 DC Characteristics: Power-Down and Supply Current PIC18F47J13 Family (Industrial) (Continued)

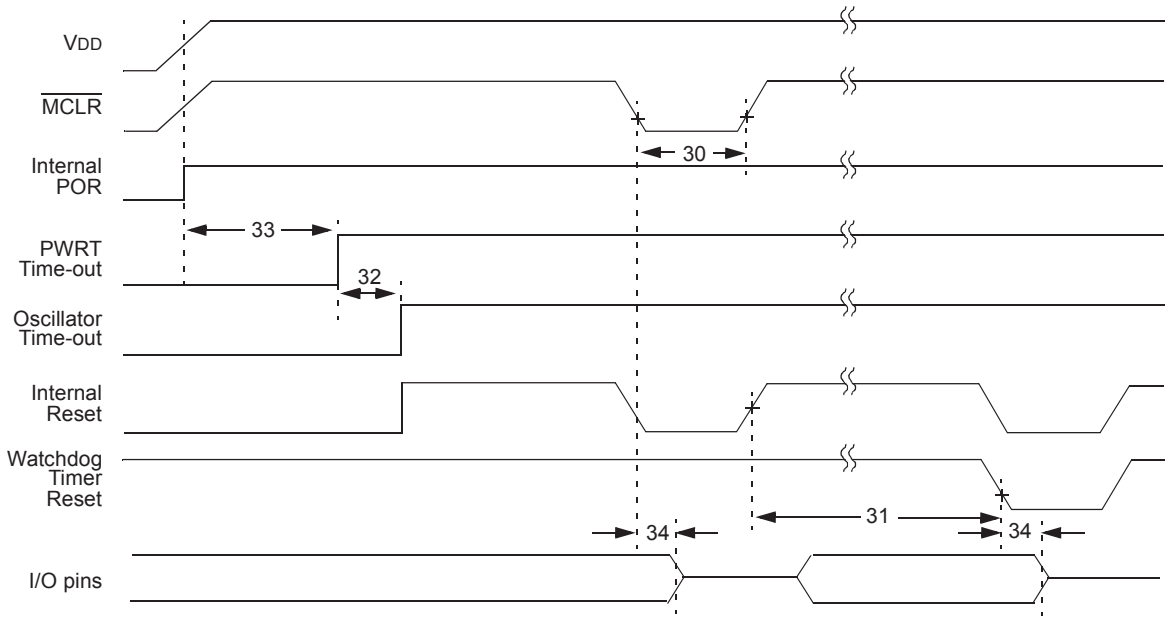
PIC18LF47J13 Family		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F47J13 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	PIC18LFXXJ13	9	45	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 32 kHz(3), **SEC_RUN** mode, SOSCSEL = 0b01
		9	45	µA	+25°C		
		12	61	µA	+85°C		
	PIC18FXXJ13	24	95	µA	-40°C	VDD = 2.15V, VDDCORE = 10 µF	
		28	95	µA	+25°C		
		35	105	µA	+85°C		
	PIC18FXXJ13	27	110	µA	-40°C	VDD = 3.3V, VDDCORE = 10 µF	
		31	110	µA	+25°C		
		35	150	µA	+85°C		
	PIC18LFXXJ13	2.5	31	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 32 kHz(3), **SEC_IDLE** mode, SOSCSEL = 0b01
		3.0	31	µA	+25°C		
		6.1	50	µA	+85°C		
	PIC18FXXJ13	19	87	µA	-40°C	VDD = 2.15V, VDDCORE = 10 µF	
		24	89	µA	+25°C		
		31	97	µA	+85°C		
	PIC18FXXJ13	21	100	µA	-40°C	VDD = 3.3V, VDDCORE = 10 µF	
		25	100	µA	+25°C		
		31	140	µA	+85°C		

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta current are disabled (such as WDT). The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD/VSS;
$\overline{MCLR}$ = VDD; WDT disabled unless otherwise specified.

3: Low-power Timer1 with standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

FIGURE 30-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 30-4 for load conditions.

TABLE 30-14: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
30	TMCL	$\overline{\text{MCLR}}$ Pulse Width (low)	2	—	—	µs	
31	TWDT	Watchdog Timer Time-out Period (no postscaler)	2.67	4.0	5.53	ms	
32	TOST	Oscillator Start-up Timer Period	1024 TOSC	—	1024 TOSC	—	TOSC = OSC1 period
33	TPWRT	Power-up Timer Period	— —	500 46	— —	µs ms	F Devices LF Devices
34	TIOZ	I/O High-Impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	3 TCY + 2	µs	**(Note 1)**
36	TIRVST	Time for Internal Reference Voltage to become Stable	—	20	—	µs	
37	TLVD	High/Low-Voltage Detect Pulse Width	—	200	—	µs	
38	TCSD	CPU Start-up Time	—	200	—	µs	**(Note 2)**

Note 1: The maximum TIOZ is the lesser of (3 TCY + 2 µs) or 700 µs.

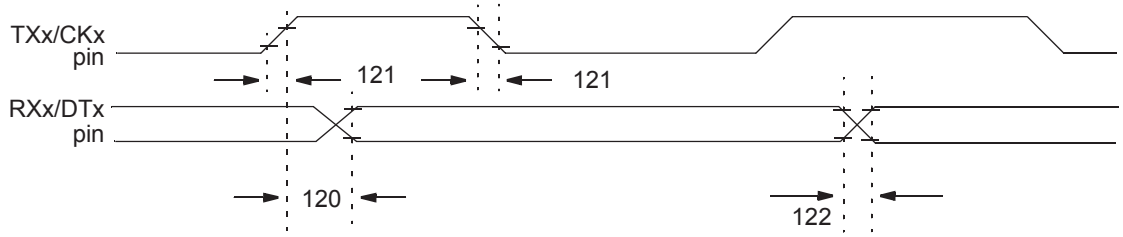
2: $\overline{\text{MCLR}}$ rising edge to code execution, assuming TPWRT (and TOST if applicable) has already expired.

TABLE 30-28: MSSPx I²C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(TOSC)(BRG + 1)	—	µs	
			400 kHz mode	2(TOSC)(BRG + 1)	—	µs	
101	TLOW	Clock Low Time	100 kHz mode	2(TOSC)(BRG + 1)	—	µs	
			400 kHz mode	2(TOSC)(BRG + 1)	—	µs	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—	µs	
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	µs	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—	µs	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	**(Note 1)**
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	µs	
			400 kHz mode	2(TOSC)(BRG + 1)	—	µs	
109	TAA	Output Valid from Clock	100 kHz mode	—	3450	ns	
			400 kHz mode	—	900	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
D102	CB	Bus Capacitive Loading		—	400	pF	

Note 1: A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.

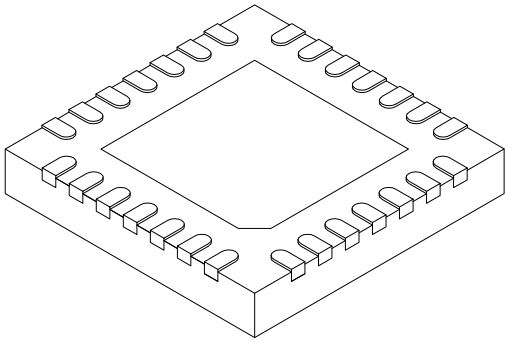
FIGURE 30-21: EUSARTx SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



Note: Refer to Figure 30-4 for load conditions.

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

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



	Units	MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105C Sheet 2 of 2