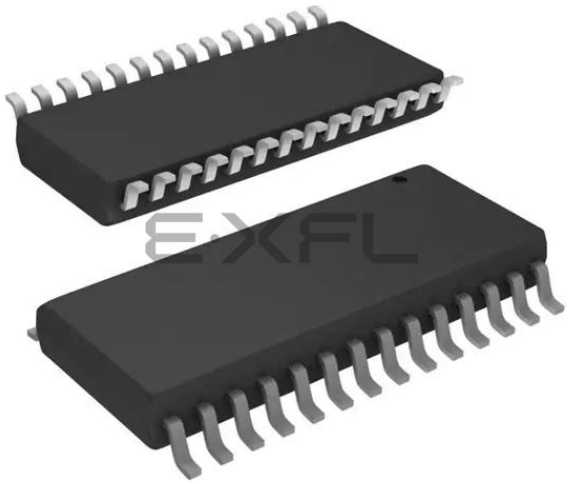


Welcome to E-XFL.COM

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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf24j11-i-so

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F24J11
- PIC18F25J11
- PIC18F26J11
- PIC18F44J11
- PIC18F45J11
- PIC18F46J11
- PIC18LF24J11
- PIC18LF25J11
- PIC18LF26J11
- PIC18LF44J11
- PIC18LF45J11
- PIC18LF46J11

1.1 Core Features

1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F46J11 family incorporate a range of features that can significantly reduce power consumption during operation. Key features are:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal RC oscillator, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operational requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the users to incorporate power-saving ideas into their application's software design.

1.1.2 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F46J11 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes using crystals or ceramic resonators.
- Two External Clock modes offering the option of a divide-by-4 clock output.
- An internal oscillator block, which provides an 8 MHz clock and an INTRC source (approximately 31 kHz, stable over temperature and V_{DD}), as well as a range of six user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of eight clock frequencies. This option frees an oscillator pin for use as an additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to the high-speed crystal, and external and internal oscillators, providing a clock speed up to 48 MHz.

The internal oscillator block provides a stable reference source that gives the PIC18F46J11 family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset (POR), or wake-up from Sleep mode, until the primary clock source is available.

1.1.3 EXPANDED MEMORY

The PIC18F46J11 family provides ample room for application code, from 16 Kbytes to 64 Kbytes of code space. The Flash cells for program memory are rated to last in excess of 10000 erase/write cycles. Data retention without refresh is conservatively estimated to be greater than 20 years.

The Flash program memory is readable and writable during normal operation. The PIC18F46J11 family also provides plenty of room for dynamic application data with up to 3.8 Kbytes of data RAM.

NOTES:

2.6 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-5. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- *AN826,* "*Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- *AN849, "Basic PICmicro® Oscillator Design"*
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*

2.7 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

FIGURE 2-5: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT

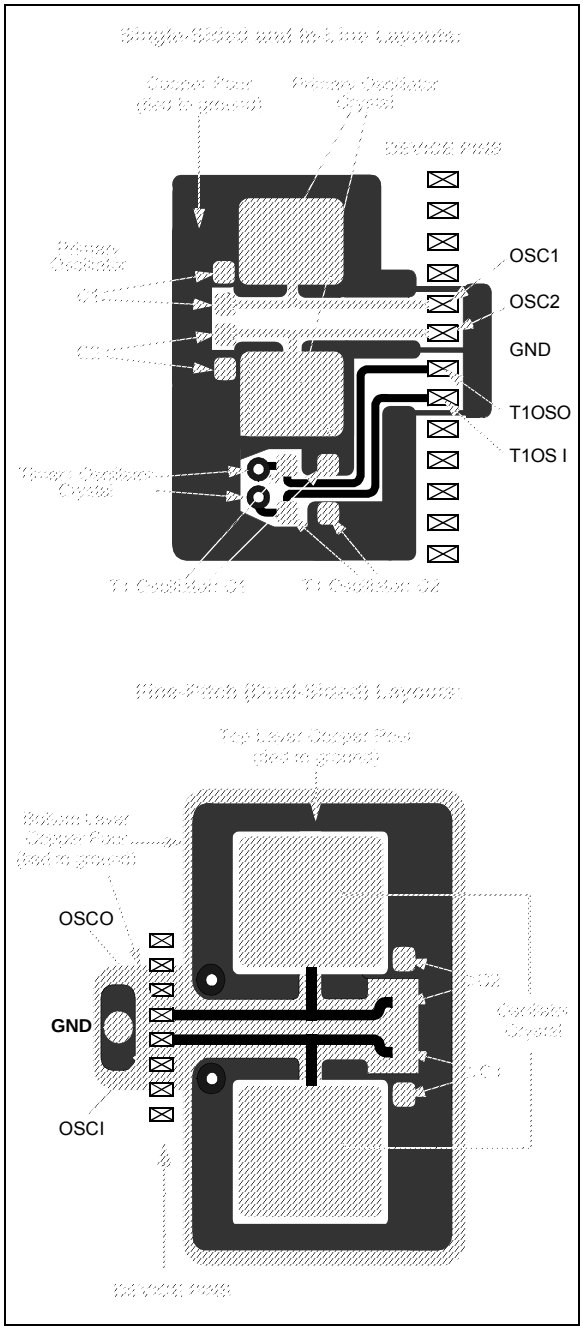


TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
PSTR1CON	PIC18F2XJ11	PIC18F4XJ11	00-0 0001	00-0 0001	uu-u uuuu
ECCP1AS	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
ECCP1DEL	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
CCPR1H	PIC18F2XJ11	PIC18F4XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F2XJ11	PIC18F4XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PSTR2CON	PIC18F2XJ11	PIC18F4XJ11	00-0 0001	00-0 0001	uu-u uuuu
ECCP2AS	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
CCPR2H	PIC18F2XJ11	PIC18F4XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F2XJ11	PIC18F4XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
CTMUCONH	PIC18F2XJ11	PIC18F4XJ11	0-00 000-	0-00 000-	u-uu uuu-
CTMUCONL	PIC18F2XJ11	PIC18F4XJ11	0000 00xx	0000 00xx	uuuu uuuu
CTMUICON	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
SPBRG1	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
TXSTA1	PIC18F2XJ11	PIC18F4XJ11	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
SPBRG2	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
TXSTA2	PIC18F2XJ11	PIC18F4XJ11	0000 0010	0000 0010	uuuu uuuu
EECON2	PIC18F2XJ11	PIC18F4XJ11	---- ----	---- ----	---- ----
EECON1	PIC18F2XJ11	PIC18F4XJ11	--00 x00-	--00 q00-	--00 u00-
IPR3	PIC18F2XJ11	PIC18F4XJ11	1111 1111	1111 1111	uuuu uuuu
PIR3	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu(3)
PIE3	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
IPR2	PIC18F2XJ11	PIC18F4XJ11	111- 1111	111- 1111	uuu- uuuu
PIR2	PIC18F2XJ11	PIC18F4XJ11	000- 0000	000- 0000	uuu- uuuu(3)
PIE2	PIC18F2XJ11	PIC18F4XJ11	000- 0000	000- 0000	uuu- 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 Reset value for specific condition.

5: Not implemented for PIC18F2XJ11 devices.

6: Not implemented on "LF" devices.

6.3 Data Memory Organization

Note: The operation of some aspects of data memory is changed when the PIC18 extended instruction set is enabled. See **Section 6.6 "Data Memory and the Extended Instruction Set"** for more information.

The data memory in PIC18 devices is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. The memory space is divided into as many as 16 banks that contain 256 bytes each. The PIC18F46J11 family implements all available banks and provides 3.8 Kbytes of data memory available to the user. Figure 6-6 provides the data memory organization for the devices.

The data memory contains Special Function Registers (SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. Any read of an unimplemented location will read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed Addressing modes. Addressing modes are discussed later in this section.

To ensure that commonly used registers (select SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to select SFRs and the lower portion of GPR Bank 0 without using the BSR. **Section 6.3.2 "Access Bank"** provides a detailed description of the Access RAM.

6.3.1 BANK SELECT REGISTER

Large areas of data memory require an efficient addressing scheme to make rapid access to any address possible. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes. Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit low-order address and a 4-bit Bank Pointer.

Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the 4 MSbs of a location's address; the instruction itself includes the 8 LSbs. Only the four lower bits of the BSR are implemented (BSR<3:0>). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the MOVLB instruction.

The value of the BSR indicates the bank in data memory. The 8 bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is illustrated in Figure 6-7.

Since, up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to an 8-bit address of F9h while the BSR is 0Fh, will end up resetting the PC.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 6-6 indicates which banks are implemented.

In the core PIC18 instruction set, only the MOVFF instruction fully specifies the 12-bit address of the source and target registers. This instruction ignores the BSR completely when it executes. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.

TABLE 10-5: PORTB I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB4/PMA1/ KBI0/RP7	RB4	0	O	DIG	LATB<4> data output; not affected by analog input.
		1	I	TTL	PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled.(1)
	PMA1(3)	0	O	DIG	Parallel Master Port address.
		1	I	ST/TTL	Parallel Slave Port address input.
	KBI0	1	I	TTL	Interrupt-on-change pin.
	RP7	1	I	ST	Remappable peripheral pin 7 input.
		0	O	DIG	Remappable peripheral pin 7 output.
RB5/PMA0/ KBI1/RP8	RB5	0	O	DIG	LATB<5> data output.
		1	I	TTL	PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	PMA0(3)	0	O	DIG	Parallel Master Port address.
		1	I	ST/TTL	Parallel Slave Port address input.
	KBI1	1	I	TTL	Interrupt-on-change pin.
	RP8	1	I	ST	Remappable peripheral pin 8 input.
		0	O	DIG	Remappable peripheral pin 8 output.
RB6/KBI2/ PGC/RP9	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI2	1	I	TTL	Interrupt-on-change pin.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation.(2)
	RP9	1	I	ST	Remappable peripheral pin 9 input.
		0	O	DIG	Remappable peripheral pin 9 output.
RB7/KBI3/ PGD/RP10	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI3	1	I	TTL	Interrupt-on-change pin.
	PGD	x	O	DIG	Serial execution data output for ICSP and ICD operation.(2)
		x	I	ST	Serial execution data input for ICSP and ICD operation.(2)
	RP10	1	I	ST	Remappable peripheral pin 10 input.
		0	O	ST	Remappable peripheral pin 10 output.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: Pins are configured as analog inputs by default on POR. Using these pins for digital inputs requires setting the appropriate bits in ANCON1 first.

2: All other pin functions are disabled when ICSP™ or ICD are enabled.

3: This bit is not available on 28-pin devices.

10.7.6 PERIPHERAL PIN SELECT REGISTERS

The PIC18F46J11 family of devices implements a total of 37 registers for remappable peripheral configuration of 44-pin devices. The 28-pin devices have 31 registers for remappable peripheral configuration.

Note: Input and output register values can only be changed if PPS<IOLOCK> = 0. See Example 10-7 for a specific command sequence.

REGISTER 10-5: PPSCON: PERIPHERAL PIN SELECT INPUT REGISTER 0 (BANKED EFFh)(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IOLOCK
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 **Unimplemented:** Read as '0'

bit 0 **IOLOCK:** I/O Lock Enable bit

1 = I/O lock active, RPORx and RPINRx registers are write-protected
0 = I/O lock not active, pin configurations can be changed

Note 1: Register values can only be changed if PPSCON<IOLOCK> = 0.

11.3.5 CHIP SELECT FEATURES

One chip select line, PMCS, is available for the Master modes of the PMP. The chip select line is multiplexed with the second Most Significant bit (MSb) of the address bus (PMADDRH<6>). When configured for chip select, the PMADDRH<7:6> bits are not included in any address auto-increment/decrement. The function of the chip select signal is configured using the chip select function bits (PMCONL<7:6>).

11.3.6 AUTO-INCREMENT/DECREMENT

While the module is operating in one of the Master modes, the INCMx bits (PMMODEH<4:3>) control the behavior of the address value. The address can be made to automatically increment or decrement after each read and write operation. The address increments once each operation is completed and the BUSY bit goes to '0'. If the chip select signals are disabled and configured as address bits, the bits will participate in the increment and decrement operations; otherwise, the CS1 bit values will be unaffected.

11.3.7 WAIT STATES

In Master mode, the user has control over the duration of the read, write and address cycles by configuring the module Wait states. Three portions of the cycle, the beginning, middle and end, are configured using the corresponding WAITBx, WAITMx and WAITEx bits in the PMMODEL register.

The WAITBx bits (PMMODEL<7:6>) set the number of Wait cycles for the data setup prior to the PMRD/PMWT strobe in Mode 10, or prior to the PMENB strobe in Mode 11. The WAITMx bits (PMMODEL<5:2>) set the number of Wait cycles for the PMRD/PMWT strobe in Mode 10, or for the PMENB strobe in Mode 11. When this Wait state setting is '0', then WAITB and WAITE have no effect. The WAITE bits (PMMODEL<1:0>) define the number of Wait cycles for the data hold time after the PMRD/PMWT strobe in Mode 10, or after the PMENB strobe in Mode 11.

11.3.8 READ OPERATION

To perform a read on the PMP, the user reads the PMDIN1L register. This causes the PMP to output the desired values on the chip select lines and the address bus. Then the read line (PMRD) is strobed. The read data is placed into the PMDIN1L register.

If the 16-bit mode is enabled (MODE16 = 1), the read of the low byte of the PMDIN1L register will initiate two bus reads. The first read data byte is placed into the PMDIN1L register, and the second read data is placed into the PMDIN1H.

Note that the read data obtained from the PMDIN1L register is actually the read value from the previous read operation. Hence, the first user read will be a dummy read to initiate the first bus read and fill the read register. Also, the requested read value will not be ready until after the BUSY bit is observed low. Thus, in a back-to-back read operation, the data read from the register will be the same for both reads. The next read of the register will yield the new value.

11.3.9 WRITE OPERATION

To perform a write onto the parallel bus, the user writes to the PMDIN1L register. This causes the module to first output the desired values on the chip select lines and the address bus. The write data from the PMDIN1L register is placed onto the PMD<7:0> data bus. Then the write line (PMWR) is strobed. If the 16-bit mode is enabled (MODE16 = 1), the write to the PMDIN1L register will initiate two bus writes. The first write will consist of the data contained in PMDIN1L and the second write will contain the PMDIN1H.

11.3.10 PARALLEL MASTER PORT STATUS

11.3.10.1 The BUSY Bit

In addition to the PMP interrupt, a BUSY bit is provided to indicate the status of the module. This bit is used only in Master mode. While any read or write operation is in progress, the BUSY bit is set for all but the very last CPU cycle of the operation. In effect, if a single-cycle read or write operation is requested, the BUSY bit will never be active. This allows back-to-back transfers. While the bit is set, any request by the user to initiate a new operation will be ignored (i.e., writing or reading the lower byte of the PMDIN1L register will neither initiate a read nor a write).

11.3.10.2 Interrupts

When the PMP module interrupt is enabled for Master mode, the module will interrupt on every completed read or write cycle; otherwise, the BUSY bit is available to query the status of the module.

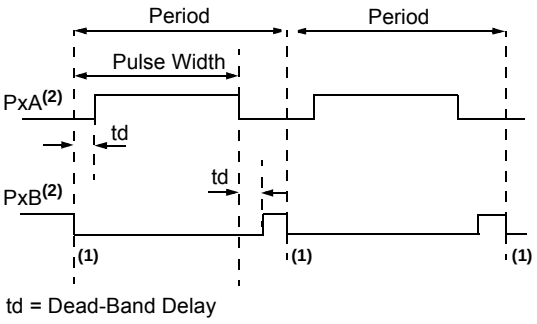
18.5.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the PxA pin, while the complementary PWM output signal is output on the PxB pin (see Figure 18-8). This mode can be used for half-bridge applications, as shown in Figure 18-9, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of the PxDC<6:0> bits of the ECCPxDEL register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 18.5.6 "Programmable Dead-Band Delay Mode"** for more details of the dead-band delay operations.

Since the PxA and PxB outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure PxA and PxB as outputs.

FIGURE 18-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT



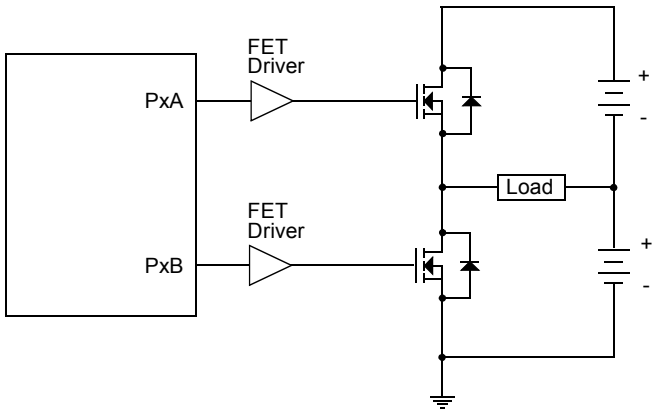
td = Dead-Band Delay

Note 1: At this time, the TMR2 register is equal to the PR2 register.

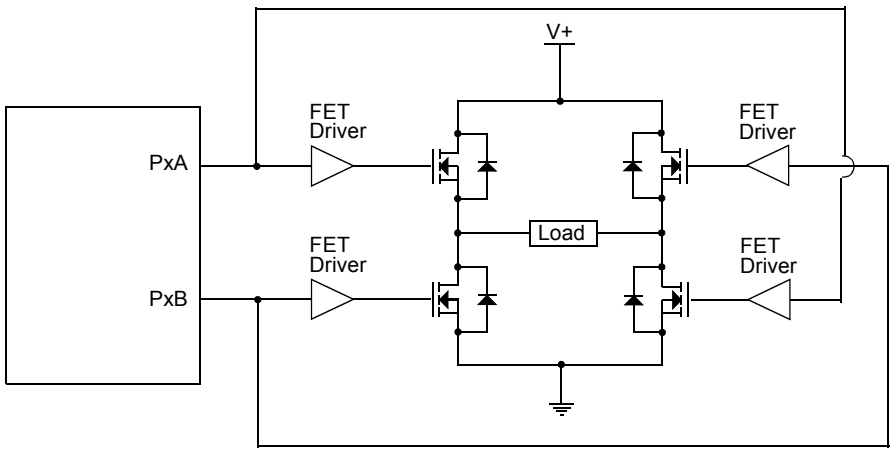
2: Output signals are shown as active-high.

FIGURE 18-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS

Standard Half-Bridge Circuit ("Push-Pull")



Half-Bridge Output Driving a Full-Bridge Circuit



19.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices include serial EEPROMs, shift registers, display drivers and A/D Converters.

19.1 Master SSP (MSSP) Module Overview

The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 5-bit and 7-bit address masking (with address masking for both 10-bit and 7-bit addressing)

All members of the PIC18F46J11 family have two MSSP modules, designated as MSSP1 and MSSP2. The modules operate independently:

- PIC18F4XJ11 devices – Both modules can be configured for either I^2C or SPI communication
- PIC18F2XJ11 devices:
  - MSSP1 can be used for either I^2C or SPI communication
  - MSSP2 can be used only for SPI communication

All of the MSSP1 module-related SPI and I^2C I/O functions are hard-mapped to specific I/O pins.

For MSSP2 functions:

- SPI I/O functions (SDO2, SDI2, SCK2 and $\overline{SS2}$) are all routed through the Peripheral Pin Select (PPS) module.

  These functions may be configured to use any of the RPn remappable pins, as described in **Section 10.7 “Peripheral Pin Select (PPS)”**.
- I^2C functions (SCL2 and SDA2) have fixed pin locations.

On all PIC18F46J11 family devices, the SPI DMA capability can only be used in conjunction with MSSP2. The SPI DMA feature is described in **Section 19.4 “SPI DMA Module”**.

Note:	Throughout this section, generic references to an MSSP module in any of its operating modes may be interpreted as being equally applicable to MSSP1 or MSSP2. Register names and module I/O signals use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module when required. Control bit names are not individuated.

19.3.9 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full-power mode. In the case of Sleep mode, all clocks are halted.

In Idle modes, a clock is provided to the peripherals. That clock can be from the primary clock source, the secondary clock (Timer1 oscillator) or the INTOSC source. See **Section 3.3 "Clock Sources and Oscillator Switching"** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set, and if enabled, will wake the device.

19.3.10 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

19.3.11 BUS MODE COMPATIBILITY

Table 19-1 provides the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 19-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit, which controls when the data is sampled.

19.3.12 SPI CLOCK SPEED AND MODULE INTERACTIONS

Because MSSP1 and MSSP2 are independent modules, they can operate simultaneously at different data rates. Setting the SSPM<3:0> bits of the SSPxCON1 register determines the rate for the corresponding module.

An exception is when both modules use Timer2 as a time base in Master mode. In this instance, any changes to the Timer2 module's operation will affect both MSSP modules equally. If different bit rates are required for each module, the user should select one of the other three time base options for one of the modules.

19.4.4.3 DMABCH and DMABCL

The DMABCH and DMABCL register pair forms a 10-bit Byte Count register, which is used by the SPI DMA module to send/receive up to 1,024 bytes for each DMA transaction. When the DMA module is actively running (DMAEN = 1), the DMA Byte Count register decrements after each byte is transmitted/received. The DMA transaction will halt and the DMAEN bit will be automatically cleared by hardware after the last byte has completed. After a DMA transaction is complete, the DMABC register will read 0x000.

Prior to initiating a DMA transaction by setting the DMAEN bit, user firmware should load the appropriate value into the DMABCH/DMABCL registers. The DMABC is a "base zero" counter, so the actual number of bytes which will be transmitted follows in Equation 19-1.

For example, if user firmware wants to transmit 7 bytes in one transaction, DMABC should be loaded with 006h. Similarly, if user firmware wishes to transmit 1,024 bytes, DMABC should be loaded with 3FFh.

EQUATION 19-1: BYTES TRANSMITTED FOR A GIVEN DMABC

$$Bytes_{XMIT} \equiv (DMABC + 1)$$

19.4.4.4 TXADDRH and TXADDRL

The TXADDRH and TXADDRL registers pair together to form a 12-bit Transmit Source Address Pointer register. In modes that use TXADDR (Full-Duplex and Half-Duplex Transmit), the TXADDR will be incremented after each byte is transmitted. Transmitted data bytes will be taken from the memory location pointed to by the TXADDR register. The contents of the memory locations pointed to by TXADDR will not be modified by the DMA module during a transmission.

The SPI DMA module can read from and transmit data from all general purpose memory on the device. The SPI DMA module cannot be used to read from the Special Function Registers (SFRs) contained in banks 14 and 15.

19.4.4.5 RXADDRH and RXADDRL

The RXADDRH and RXADDRL register pair together to form a 12-bit Receive Destination Address Pointer. In modes that use RXADDR (Full-Duplex and Half-Duplex Receive), the RXADDR register will be incremented after each byte is received. Received data bytes will be stored at the memory location pointed to by the RXADDR register.

The SPI DMA module can write received data to all general purpose memory on the device. The SPI DMA module cannot be used to modify the Special Function Registers contained in banks 14 and 15.

19.4.5 INTERRUPTS

The SPI DMA module alters the behavior of the SSP2IF interrupt flag. In normal/non-DMA modes, the SSP2IF is set once after every single byte is transmitted/received through the MSSP2 module. When MSSP2 is used with the SPI DMA module, the SSP2IF interrupt flag will be set according to the user-selected INTLVL<3:0> value specified in the DMACON2 register. The SSP2IF interrupt condition will also be generated once the SPI DMA transaction has fully completed, and the DMAEN bit has been cleared by hardware.

The SSP2IF flag becomes set once the DMA byte count value indicates that the specified INTLVL has been reached. For example, if DMACON2<3:0> = 0101 (16 bytes remaining), the SSP2IF interrupt flag will become set once DMABC reaches 00Fh. If user firmware then clears the SSP2IF interrupt flag, the flag will not be set again by the hardware until after all bytes have been fully transmitted and the DMA transaction is complete.

Note:	User firmware may modify the INTLVL bits while a DMA transaction is in progress (DMAEN = 1). If an INTLVL value is selected which is higher than the actual remaining number of bytes (indicated by DMABC + 1), the SSP2IF interrupt flag will immediately become set.

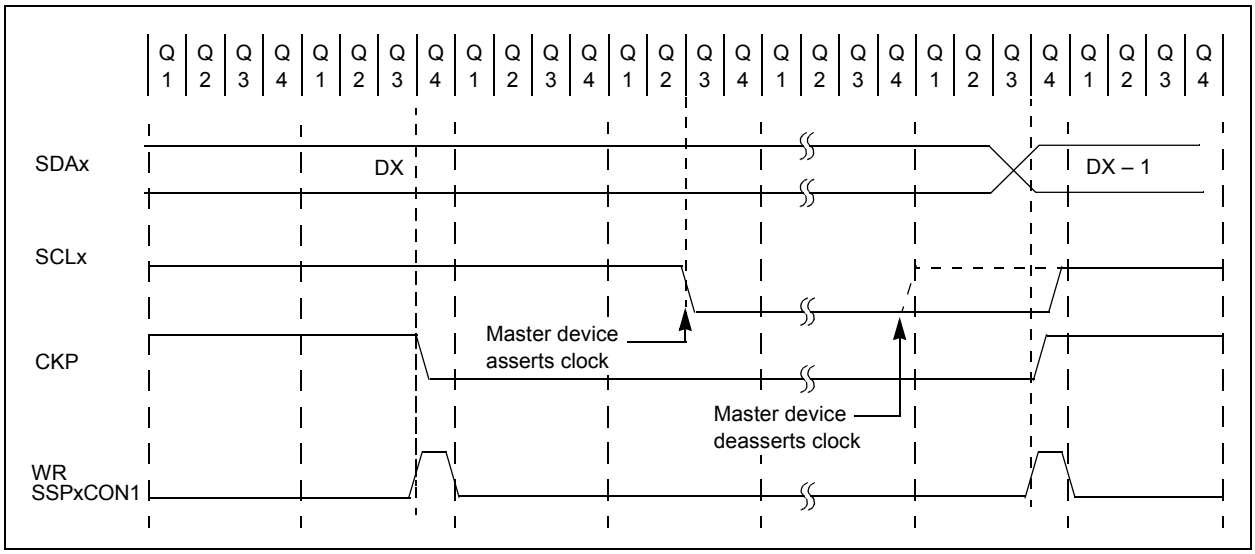
For example, if DMABC = 00Fh (implying 16 bytes are remaining) and user firmware writes '1111' to INTLVL<3:0> (interrupt when 576 bytes remaining), the SSP2IF interrupt flag will immediately become set. If user firmware clears this interrupt flag, a new interrupt condition will not be generated until either: user firmware again writes INTLVL with an interrupt level higher than the actual remaining level, or the DMA transaction completes and the DMAEN bit is cleared.

Note:	If the INTLVL bits are modified while a DMA transaction is in progress, care should be taken to avoid inadvertently changing the DLYCYC<3:0> value.

19.5.4.5 Clock Synchronization and CKP bit

When the CKP bit is cleared, the SCLx output is forced to '0'. However, clearing the CKP bit will not assert the SCLx output low until the SCLx output is already sampled low. Therefore, the CKP bit will not assert the SCLx line until an external I^2C master device has already asserted the SCLx line. The SCLx output will remain low until the CKP bit is set and all other devices on the I^2C bus have deasserted SCLx. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCLx (see Figure 19-14).

FIGURE 19-14: CLOCK SYNCHRONIZATION TIMING



20.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any Idle mode and bit, SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREGx register. If the RCxIE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit, RCxIE.
3. If 9-bit reception is desired, set bit, RX9.
4. To enable reception, set enable bit, CREN.
5. Flag bit, RCxIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCxIE, was set.
6. Read the RCSTAx register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREGx register.
8. If any error occurred, clear the error by clearing bit, CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 20-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF(1)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE(1)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP(1)	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	72
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	72
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	72
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
RCREGx	EUSARTx Receive Register								72
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	72
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								73
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								72

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

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

NOTES:

REGISTER 26-10: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F46J11 FAMILY DEVICES (BYTE ADDRESS 3FFFFFh)

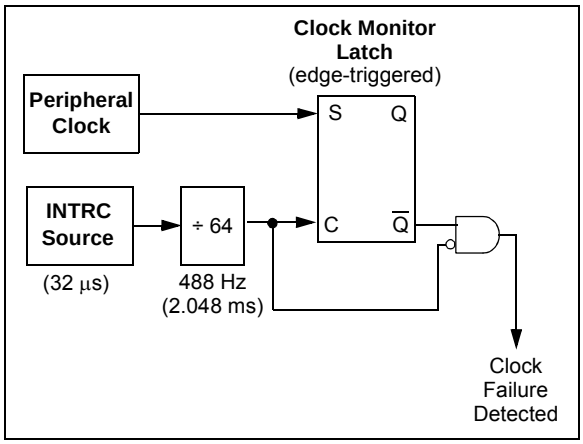
R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

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

bit 7-0 **DEV<10:3>:** Device ID bits
These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

DEV<10:3> (DEVID2<7:0>)	DEV<2:0> (DEVID1<7:5>)	Device
0100 1110	001	PIC18F46J11
0100 1110	000	PIC18F45J11
0100 1101	111	PIC18F44J11
0100 1101	110	PIC18F26J11
0100 1101	101	PIC18F25J11
0100 1101	100	PIC18F24J11
0100 1110	111	PIC18LF46J11
0100 1110	110	PIC18LF45J11
0100 1110	101	PIC18LF44J11
0100 1110	100	PIC18LF26J11
0100 1110	011	PIC18LF25J11
0100 1110	010	PIC18LF24J11

FIGURE 26-4: 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 26-5), 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 26.4.1 "Special Considerations for Using Two-Speed Start-up"** for more details.

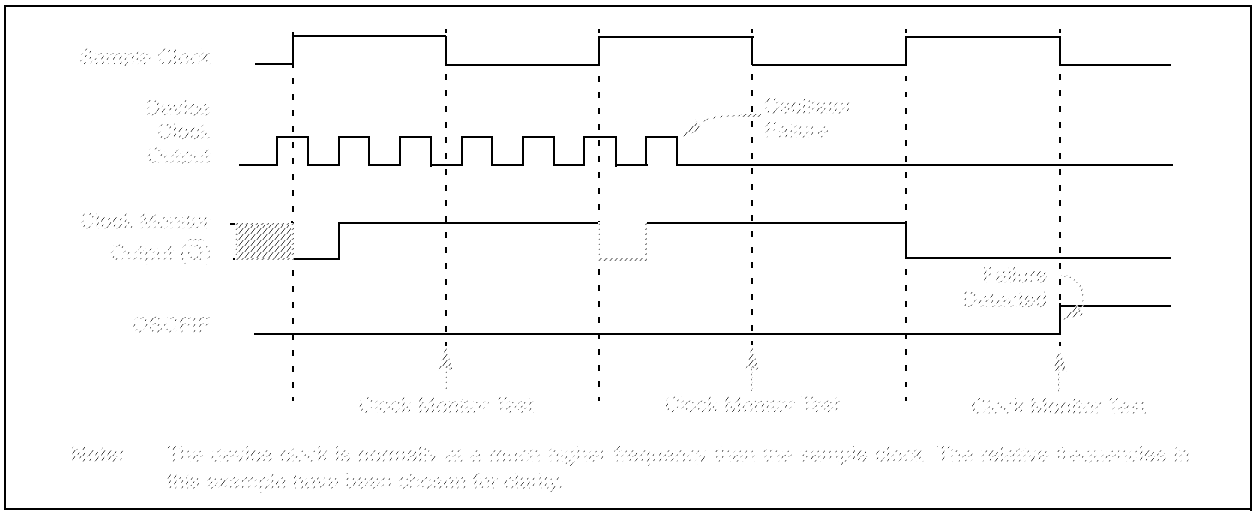
The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

26.5.1 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 26-5: FSCM TIMING DIAGRAM



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 $f \le 95$ (5Fh). See **Section 27.2.3 “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 LATC, 4, 0`

Before Instruction:
LATC = 0111 0101 [75h]
After Instruction:
LATC = 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 two-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

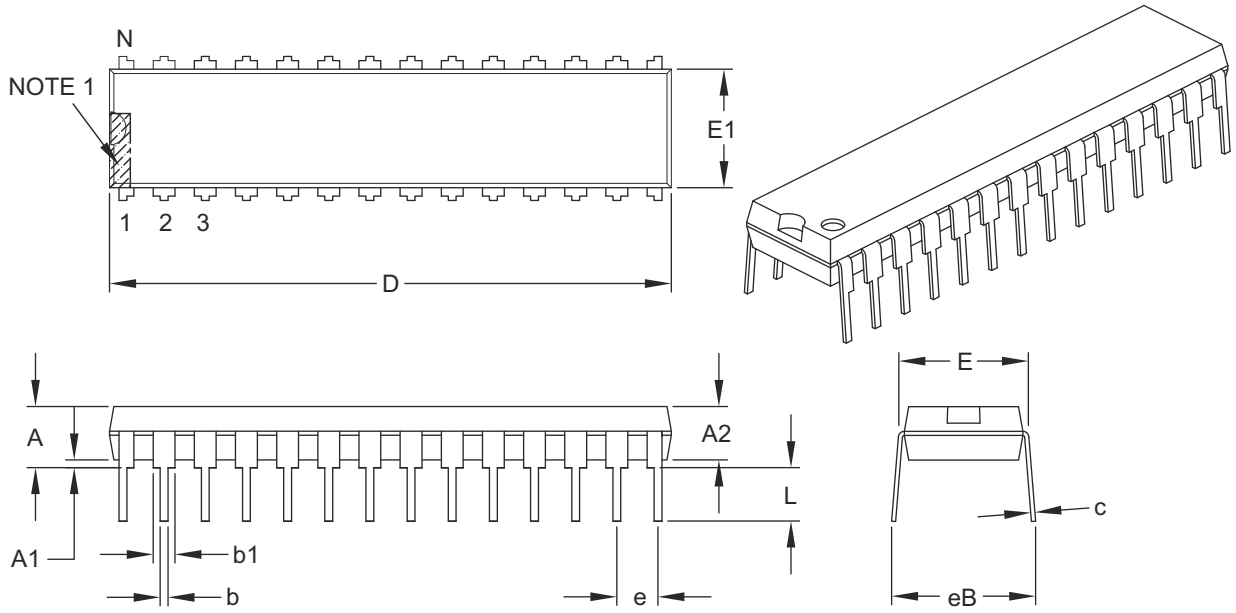
Example:
```
HERE    BOV  Jump
```
Before Instruction
PC = address (HERE)
After Instruction
If Overflow = 1;
PC = address (Jump)
If Overflow = 0;
PC = address (HERE + 2)

30.2 Package Details

The following sections give the technical details of the packages.

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-070B