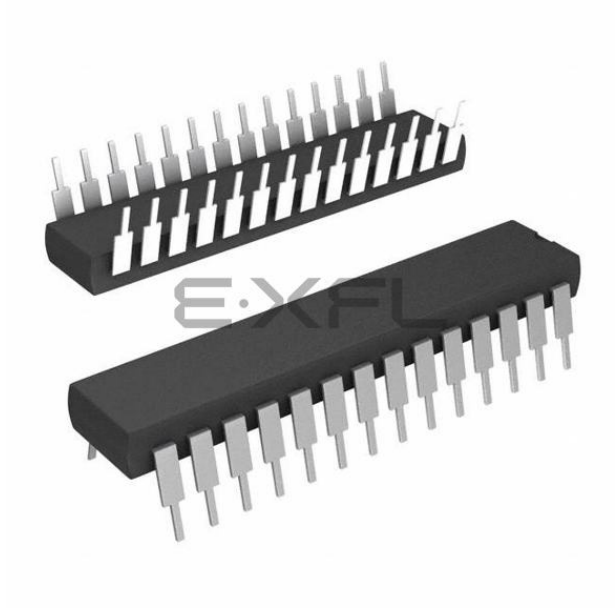


Welcome to E-XFL.COM

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	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	48KB (24K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf2585-i-sp

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 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, 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.

3.0 POWER MANAGED MODES

PIC18F2585/2680/4585/4680 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).

There are three categories of power managed modes:

- Run modes
- Idle modes
- Sleep mode

These categories define which portions of the device are clocked and sometimes, what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block); the Sleep mode does not use a clock source.

The power managed modes include several power saving features offered on previous PIC® devices. One is the clock switching feature, offered in other PIC18 devices, allowing the controller to use the Timer1 oscillator in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC devices, where all device clocks are stopped.

3.1 Selecting Power Managed Modes

Selecting a power managed mode requires two decisions: if the CPU is to be clocked or not and the selection of a clock source. The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS1:SCS0 bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

3.1.1 CLOCK SOURCES

The SCS1:SCS0 bits allow the selection of one of three clock sources for power managed modes. They are:

- the primary clock, as defined by the FOSC3:FOSC0 Configuration bits
- the secondary clock (the Timer1 oscillator)
- the internal oscillator block (for RC modes)

3.1.2 ENTERING POWER MANAGED MODES

Switching from one power managed mode to another begins by loading the OSCCON register. The SCS1:SCS0 bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. These are discussed in **Section 3.1.3 "Clock Transitions And Status Indicators"** and subsequent sections.

Entry to the Power Managed Idle or Sleep modes is triggered by the execution of a `SLEEP` instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a `SLEEP` instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a `SLEEP` instruction to switch to the desired mode.

TABLE 3-1: POWER MANAGED MODES

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN<7>(1)	SCS1:SCS0<1:0>	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	00	Clocked	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC, INTRC(2): This is the normal full power execution mode.
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – Timer1 Oscillator
RC_RUN	N/A	1x	Clocked	Clocked	Internal Oscillator Block(2)
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – Timer1 Oscillator
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block(2)

Note 1: IDLEN reflects its value when the `SLEEP` instruction is executed.

2: Includes INTOSC and INTOSC postscaler, as well as the INTRC source.

3.4.3 RC_IDLE MODE

In RC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator block using the INTOSC multiplexer. This mode allows for controllable power conservation during Idle periods.

From RC_RUN, this mode is entered by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, first set IDLEN, then set the SCS1 bit and execute SLEEP. Although its value is ignored, it is recommended that SCS0 also be cleared; this is to maintain software compatibility with future devices. The INTOSC multiplexer may be used to select a higher clock frequency, by modifying the IRCF bits, before executing the SLEEP instruction. When the clock source is switched to the INTOSC multiplexer, the primary oscillator is shut down and the OSTS bit is cleared.

If the IRCF bits are set to any non-zero value or the INTSRC bit is set, the INTOSC output is enabled. The IOFS bit becomes set, after the INTOSC output becomes stable, after an interval of TIOBST (parameter 39, Table 27-10). Clocks to the peripherals continue while the INTOSC source stabilizes. If the IRCF bits were previously at a non-zero value, or INTSRC was set before the SLEEP instruction was executed and the INTOSC source was already stable, the IOFS bit will remain set. If the IRCF bits and INTSRC are all clear, the INTOSC output will not be enabled, the IOFS bit will remain clear and there will be no indication of the current clock source.

When a wake event occurs, the peripherals continue to be clocked from the INTOSC multiplexer. After a delay of TCSD following the wake event, the CPU begins executing code being clocked by the INTOSC multiplexer. The IDLEN and SCS bits are not affected by the wake-up. The INTRC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

3.5 Exiting Idle and Sleep Modes

An exit from Sleep mode or any of the Idle modes is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power managed modes. The clocking subsystem actions are discussed in each of the power managed modes (see **Section 3.2 “Run Modes”, Section 3.3 “Sleep Mode”** and **Section 3.4 “Idle Modes”**).

3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode or the Sleep mode to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

On all exits from Idle or Sleep modes by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 9.0 “Interrupts”**).

A fixed delay of interval TCSD following the wake event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

3.5.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power managed mode (see **Section 3.2 “Run Modes”** and **Section 3.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 24.2 “Watchdog Timer (WDT)”**).

The WDT timer and postscaler are cleared by executing a SLEEP or CLRWDT instruction, the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifying the IRCF bits in the OSCCON register if the internal oscillator block is the device clock source.

3.5.3 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock becomes ready. At that time, the OSTS bit is set and the device begins executing code. If the internal oscillator block is the new clock source, the IOFS bit is set instead.

The exit delay time from Reset to the start of code execution depends on both the clock sources before and after the wake-up and the type of oscillator if the new clock source is the primary clock. Exit delays are summarized in Table 3-2.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see **Section 24.3 “Two-Speed Start-up”**) or Fail-Safe Clock Monitor (see **Section 24.4 “Fail-Safe Clock Monitor”**) is enabled, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Execution is clocked by the internal oscillator block until either the primary clock becomes ready or a power managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power managed modes do not invoke the OST at all. There are two cases:

- PRI_IDLE mode where the primary clock source is not stopped; and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval TCSD following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

TABLE 3-2: EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)

Clock Source Before Wake-up	Clock Source After Wake-up	Exit Delay	Clock Ready Status Bit (OSCCON)
Primary Device Clock (PRI_IDLE mode)	LP, XT, HS	T_{CSD}(2)	OSTS
	HSPLL		
	EC, RC		
	INTRC(1)		—
	INTOSC(3)		IOFS
T1OSC or INTRC(1)	LP, XT, HS	T_{OST}(4)	OSTS
	HSPLL	$T_{OST} + t_{rc}$(4)	
	EC, RC	T_{CSD}(2)	
	INTRC(1)		—
	INTOSC(2)	T_{IOBST}(5)	IOFS
INTOSC(3)	LP, XT, HS	T_{OST}(5)	OSTS
	HSPLL	$T_{OST} + t_{rc}$(4)	
	EC, RC	T_{CSD}(2)	
	INTRC(1)		—
	INTOSC(2)	None	IOFS
None (Sleep mode)	LP, XT, HS	T_{OST}(4)	OSTS
	HSPLL	$T_{OST} + t_{rc}$(4)	
	EC, RC	T_{CSD}(2)	
	INTRC(1)		—
	INTOSC(2)	T_{IOBST}(5)	IOFS

Note 1: In this instance, refers specifically to the 31 kHz INTRC clock source.

2: TCSD (parameter 38) is a required delay when waking from Sleep and all Idle modes and runs concurrently with any other required delays (see **Section 3.4 "Idle Modes"**).

3: Includes both the INTOSC 8 MHz source and postscaler derived frequencies.

4: TOST is the Oscillator Start-up Timer (parameter 32). t_{rc} is the PLL Lock-out Timer (parameter F12); it is also designated as TPLL.

5: Execution continues during TIOBST (parameter 39), the INTOSC stabilization period.

TABLE 4-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
CCPR1H	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	2585	2680	4585	4680	--00 0000	--00 0000	--uu uuuu
ECCPR1H	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
ECCPR1L	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
ECCP1CON	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
BAUDCON	2585	2680	4585	4680	01-0 0-00	01-0 0-00	--uu uuuu
ECCP1DEL	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
ECCP1AS	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
CVRCON	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
CMCON	2585	2680	4585	4680	0000 0111	0000 0111	uuuu uuuu
TMR3H	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	2585	2680	4585	4680	0000 0000	uuuu uuuu	uuuu uuuu
SPBRGH	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
SPBRG	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
RCREG	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
TXREG	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
TXSTA	2585	2680	4585	4680	0000 0010	0000 0010	uuuu uuuu
RCSTA	2585	2680	4585	4680	0000 000x	0000 000x	uuuu uuuu
EEADRH	2585	2680	4585	4680	---- --00	---- --00	---- --uu
EEADR	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
EEDATA	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
EECON2	2585	2680	4585	4680	0000 0000	0000 0000	0000 0000
EECON1	2585	2680	4585	4680	xx-0 x000	uu-0 u000	uu-0 u000
IPR3	2585	2680	4585	4680	1111 1111	1111 1111	uuuu uuuu
PIR3	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
PIE3	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
IPR2	2585	2680	4585	4680	11-1 1111	11-1 1111	uu-u uuuu
	2585	2680	4585	4680	1--1 111-	1--1 111-	u--u uuu-
PIR2	2585	2680	4585	4680	00-0 0000	00-0 0000	uu-u uuuu(1)
	2585	2680	4585	4680	0--0 000-	0--0 000-	u--u uuu-(1)

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

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: 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.

4: See Table 4-3 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: This register reads all '0's until the ECAN™ technology is set up in Mode 1 or Mode 2.

TABLE 4-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
CANCON	2585	2680	4585	4680	1000 000-	1000 000-	uuuu uuu-
CANSTAT	2585	2680	4585	4680	100- 000-	100- 000-	uuu- uuu-
RXB0D7	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D6	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D5	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D4	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D3	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D2	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D1	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D0	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0DLC	2585	2680	4585	4680	-xxx xxxx	-uuu uuuu	-uuu uuuu
RXB0EIDL	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0EIDH	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0SIDL	2585	2680	4585	4680	xxxx x-xx	uuuu u-uu	uuuu u-uu
RXB0SIDH	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0CON	2585	2680	4585	4680	000- 0000	000- 0000	uuu- uuuu
RXB1D7	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D6	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D5	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D4	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D3	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D2	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D1	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D0	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1DLC	2585	2680	4585	4680	-xxx xxxx	-uuu uuuu	-uuu uuuu
RXB1EIDL	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1EIDH	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1SIDL	2585	2680	4585	4680	xxxx x-xx	uuuu u-uu	uuuu u-uu
RXB1SIDH	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1CON	2585	2680	4585	4680	000- 0000	000- 0000	uuu- uuuu
TXB0D7	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D6	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

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: 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.

4: See Table 4-3 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: This register reads all '0's until the ECAN™ technology is set up in Mode 1 or Mode 2.

TABLE 4-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
B4EIDH(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4SIDL(6)	2585	2680	4585	4680	xxxx x-xx	uuuu u-uu	uuuu u-uu
B4SIDH(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4CON(6)	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
B3D7(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D6(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D5(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D4(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D3(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D2(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D1(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D0(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3DLC(6)	2585	2680	4585	4680	-xxx xxxx	-uuu uuuu	-uuu uuuu
B3EIDL(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3EIDH(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3SIDL(6)	2585	2680	4585	4680	xxxx x-xx	uuuu u-uu	uuuu u-uu
B3SIDH(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3CON(6)	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
B2D7(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D6(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D5(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D4(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D3(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D2(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D1(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D0(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2DLC(6)	2585	2680	4585	4680	-xxx xxxx	-uuu uuuu	-uuu uuuu
B2EIDL(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2EIDH(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2SIDL(6)	2585	2680	4585	4680	xxxx x-xx	uuuu u-uu	uuuu u-uu
B2SIDH(6)	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

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: 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.

4: See Table 4-3 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: This register reads all '0's until the ECAN™ technology is set up in Mode 1 or Mode 2.

TABLE 10-7: PORTD I/O SUMMARY

Pin Name	Function	I/O	TRIS	Buffer	Description
RD0/PSP0/ C1IN+	RD0	OUT	0	DIG	LATD<0> data output.
		IN	1	ST	PORTD<0> data input.
	PSP0	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<0> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<0> control when enabled).
	C1IN+	IN	1	ANA	Comparator 1 positive input B. Default on POR. This analog input overrides the digital input (read as clear – low level).
RD1/PSP1/ C1IN-	RD1	OUT	0	DIG	LATD<1> data output.
		IN	1	ST	PORTD<1> data input.
	PSP1	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<1> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<1> control when enabled).
	C1IN-	IN	1	ANA	Comparator 1 negative input. Default on POR. This analog input overrides the digital input (read as clear – low level).
RD2/PSP2/ C2IN+	RD2	OUT	0	DIG	LATD<2> data output.
		IN	1	ST	PORTD<2> data input.
	PSP2	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<2> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<2> control when enabled).
	C2IN+	IN	1	ANA	Comparator 2 positive input. Default on POR. This analog input overrides the digital input (read as clear – low level).
RD3/PSP3/ C2IN-	RD3	OUT	0	DIG	LATD<3> data output.
		IN	1	ST	PORTD<3> data input.
	PSP3	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<3> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<3> control when enabled).
	C2IN-	IN	1	ANA	Comparator 2 negative input. Default input on POR. This analog input overrides the digital input (read as clear – low level).
RD4/PSP4/ ECCP1/P1A	RD4	OUT	0	DIG	LATD<4> data output.
		IN	1	ST	PORTD<4> data input.
	PSP4	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<4> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<4> control when enabled).
	ECCP1	OUT	0	DIG	ECCP1 compare output.
		IN	1	ST	ECCP1 capture input.
	P1A	OUT	0	DIG	ECCP1 Enhanced PWM output, channel A.
RD5/PSP5/ P1B	RD5	OUT	0	DIG	LATD<5> data output.
		IN	1	ST	PORTD<5> data input.
	PSP5	OUT	X	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<5> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<5> control when enabled).
	P1B	OUT	0	DIG	ECCP1 Enhanced PWM output, channel B.
RD6/PSP6/ P1C	RD6	OUT	0	DIG	LATD<6> data output.
		IN	1	ST	PORTD<6> data input.
	PSP6	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<6> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<6> control when enabled).
	P1C	OUT	0	DIG	ECCP1 Enhanced PWM output, channel C.
RD7/PSP7/ P1D	RD7	OUT	0	DIG	LATD<7> data output.
		IN	1	ST	PORTD<7> data input.
	PSP7	OUT	x	DIG	Parallel Slave Port (PSP) data output (overrides the TRIS<7> control when enabled).
		IN	x	TTL	Parallel Slave Port (PSP) data input (overrides the TRIS<7> control when enabled).
	P1D	OUT	0	DIG	ECCP1 Enhanced PWM output, channel D.

Legend: PWR = Power Supply; OUT = Output; IN = Input; ANA = Analog Signal; DIG = Digital Output; ST = Schmitt Buffer Input; TTL = TTL Buffer Input

17.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge sequence enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 17-23).

17.4.12.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

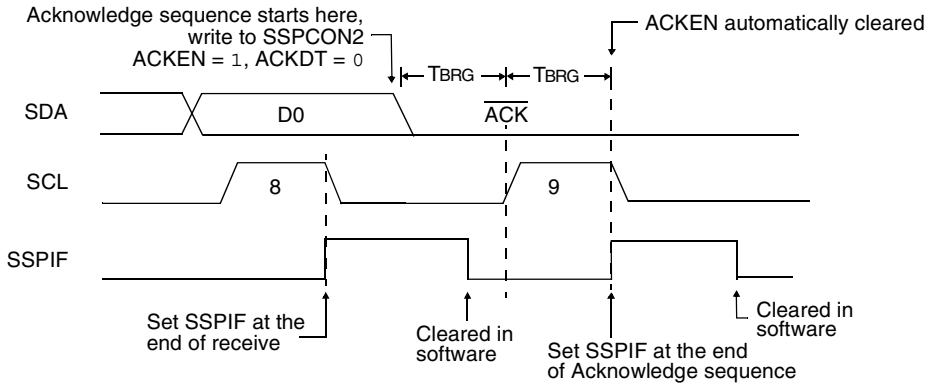
17.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 17-24).

17.4.13.1 WCOL Status Flag

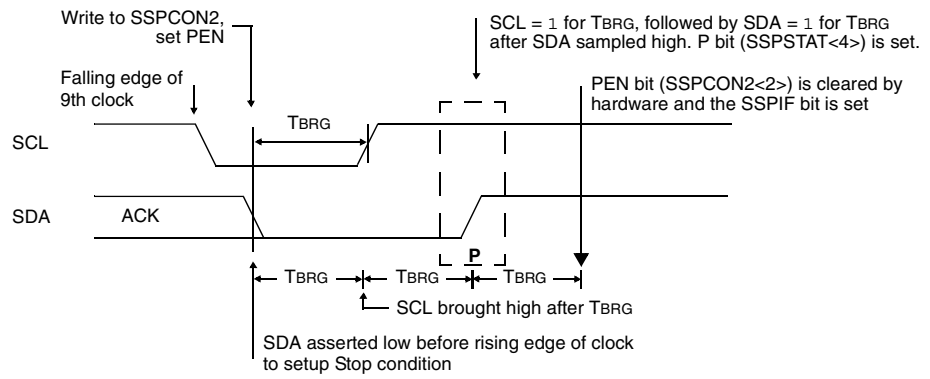
If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 17-23: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 17-24: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

18.4.2 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 RCREG register; if the RCIE 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 RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete. An interrupt will be generated if enable bit RCIE was set.
6. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG 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 18-10: 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	49
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART Receive Register								51
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART Baud Rate Generator Register High Byte								51
SPBRG	EUSART Baud Rate Generator Register Low Byte								51

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

Note 1: Reserved in PIC18F2X8X devices; always maintain these bits clear.

19.4 Operation in Power Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined in part, by the clock source and frequency while in a power managed mode.

If the A/D is expected to operate while the device is in a power managed mode, the ACQT2:ACQT0 and ADCS2:ADCS0 bits in ADCON2 should be updated in accordance with the clock source to be used in that mode. After entering the mode, an A/D acquisition or conversion may be started. Once started, the device should continue to be clocked by the same clock source until the conversion has been completed.

If desired, the device may be placed into the corresponding Idle mode during the conversion. If the device clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

Operation in the Sleep mode requires the A/D FRC clock to be selected. If bits ACQT2:ACQT0 are set to '000' and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry to Sleep mode. The IDLEN bit (OSCCON<7>) must have already been cleared prior to starting the conversion.

19.5 Configuring Analog Port Pins

The ADCON1, TRISA, TRISB and TRISE registers all configure the A/D port pins. The port pins needed as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

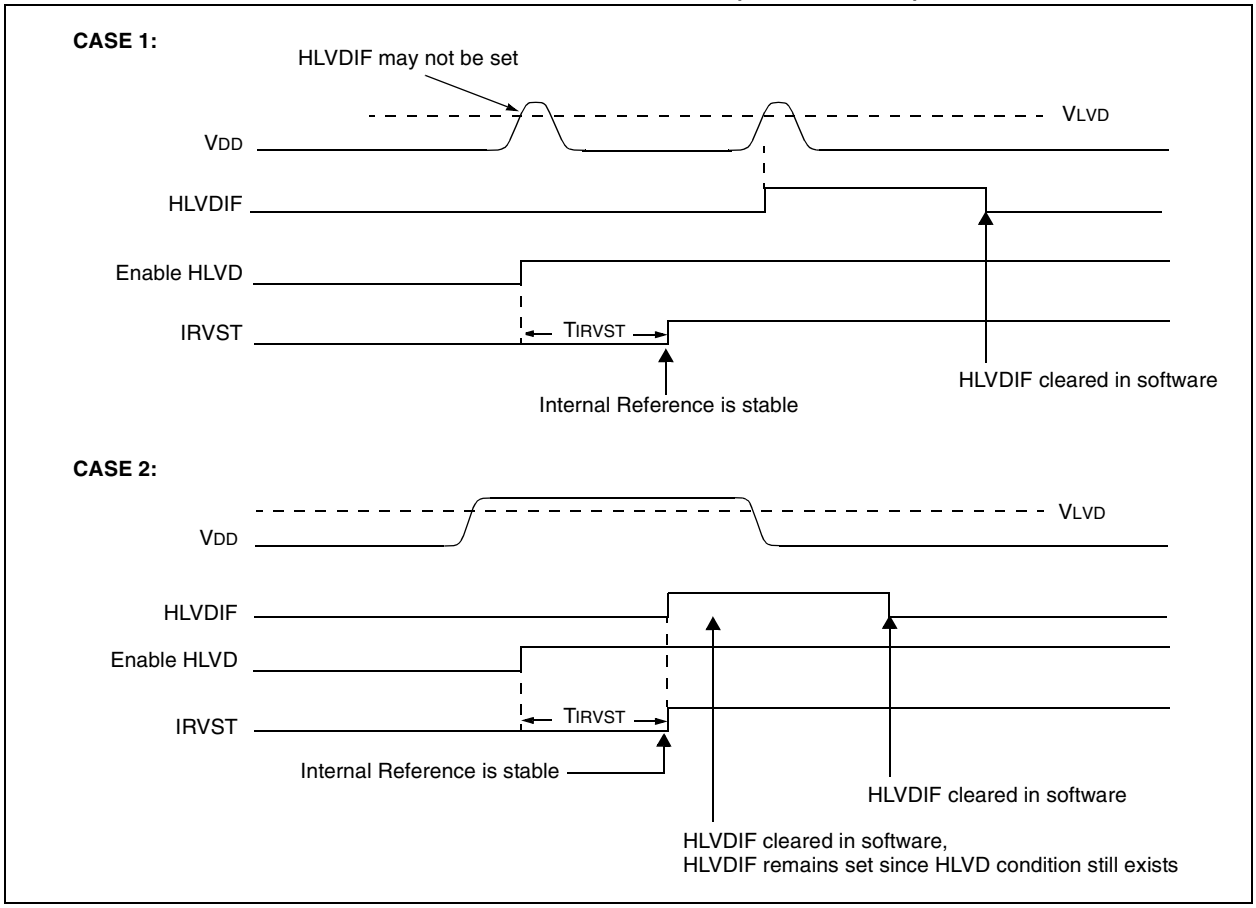
The A/D operation is independent of the state of the CHS3:CHS0 bits and the TRIS bits.

Note 1: When reading the Port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will be accurately converted.

2: Analog levels on any pin defined as a digital input may cause the digital input buffer to consume current out of the device's specification limits.

3: The PBADEN bit in Configuration Register 3H configures PORTB pins to reset as analog or digital pins by controlling how the PCFG0 bits in ADCON1 are reset.

FIGURE 22-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)

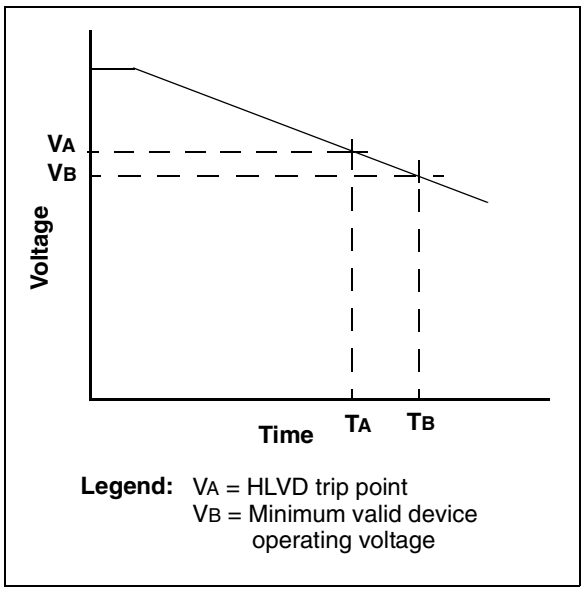


22.5 Applications

In many applications, the ability to detect a drop below, or rise above a particular threshold is desirable. For example, the HLVD module could be periodically enabled to detect Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a high-voltage detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 22-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage VA, the HLVD logic generates an interrupt at time TA. The interrupt could cause the execution of an ISR, which would allow the application to perform "house-keeping tasks" and perform a controlled shutdown before the device voltage exits the valid operating range at TB. The HLVD, thus, would give the application a time window, represented by the difference between TA and TB, to safely exit.

FIGURE 22-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION



REGISTER 23-2: CANSTAT: CAN STATUS REGISTER

Mode 0

R-1	R-0	R-0	R-0	R-0	R-0	R-0	U-0
OPMODE2(1)	OPMODE1(1)	OPMODE0(1)	—	ICODE3	ICODE2	ICODE1	—

Mode 1, 2

R-1	R-0	R-0	R-0	R-0	R-0	R-0	R-0
OPMODE2(1)	OPMODE1(1)	OPMODE0(1)	EICODE4	EICODE3	EICODE2	EICODE1	EICODE0

bit 7 bit 0

bit 7-5 **OPMODE2:OPMODE0:** Operation Mode Status bits(1)

111 = Reserved
110 = Reserved
101 = Reserved
100 = Configuration mode
011 = Listen Only mode
010 = Loopback mode
001 = Disable/Sleep mode
000 = Normal mode

bit 4 Mode 0:

Unimplemented: Read as '0'

bit 3-1 **ICODE3:ICODE1:** Interrupt Code bits

When an interrupt occurs, a prioritized coded interrupt value will be present in these bits. This code indicates the source of the interrupt. By copying ICODE3:ICODE1 to WIN2:WIN0 (Mode 0) or EICODE4:EICODE0 to EWIN4:EWIN0 (Mode 1 and 2), it is possible to select the correct buffer to map into the Access Bank area. See Example 23-2 for a code example. To simplify the description, the following table lists all five bits.

	Mode 0	Mode 1	Mode 2
No interrupt	00000	00000	00000
Error interrupt	00010	00010	00010
TXB2 interrupt	00100	00100	00100
TXB1 interrupt	00110	00110	00110
TXB0 interrupt	01000	01000	01000
RXB1 interrupt	01010	10001	-----
RXB0 interrupt	01100	10000	10000
Wake-up interrupt	00010	01110	01110
RXB0 interrupt	-----	10000	10000
RXB1 interrupt	-----	10001	10000
RX/TX B0 interrupt	-----	10010	10010
RX/TX B1 interrupt	-----	10011	10011(2)
RX/TX B2 interrupt	-----	10100	10100(2)
RX/TX B3 interrupt	-----	10101	10101(2)
RX/TX B4 interrupt	-----	10110	10110(2)
RX/TX B5 interrupt	-----	10111	10111(2)

bit 0 **Unimplemented:** Read as '0'

bit 4-0 Mode 1, 2:

EICODE4:EICODE0: Interrupt Code bits

See ICODE3:ICODE1 above.

Note 1: To achieve maximum power saving and/or able to wake-up on CAN bus activity, switch CAN module in Disable mode before putting device to Sleep.

2: If buffer is configured as receiver, EICODE bits will contain '10000' upon interrupt.

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

23.6.3 TRANSMIT PRIORITY

Transmit priority is a prioritization within the PIC18F2585/2680/4585/4680 devices of the pending transmittable messages. This is independent from and not related to any prioritization implicit in the message arbitration scheme built into the CAN protocol. Prior to sending the SOF, the priority of all buffers that are queued for transmission is compared. The transmit buffer with the highest priority will be sent first. If two buffers have the same priority setting, the buffer with the highest buffer number will be sent first. There are four levels of transmit priority. If TXP bits for a particular message buffer are set to '11', that buffer has the highest possible priority. If TXP bits for a particular message buffer are set to '00', that buffer has the lowest possible priority.

FIGURE 23-2: TRANSMIT BUFFERS

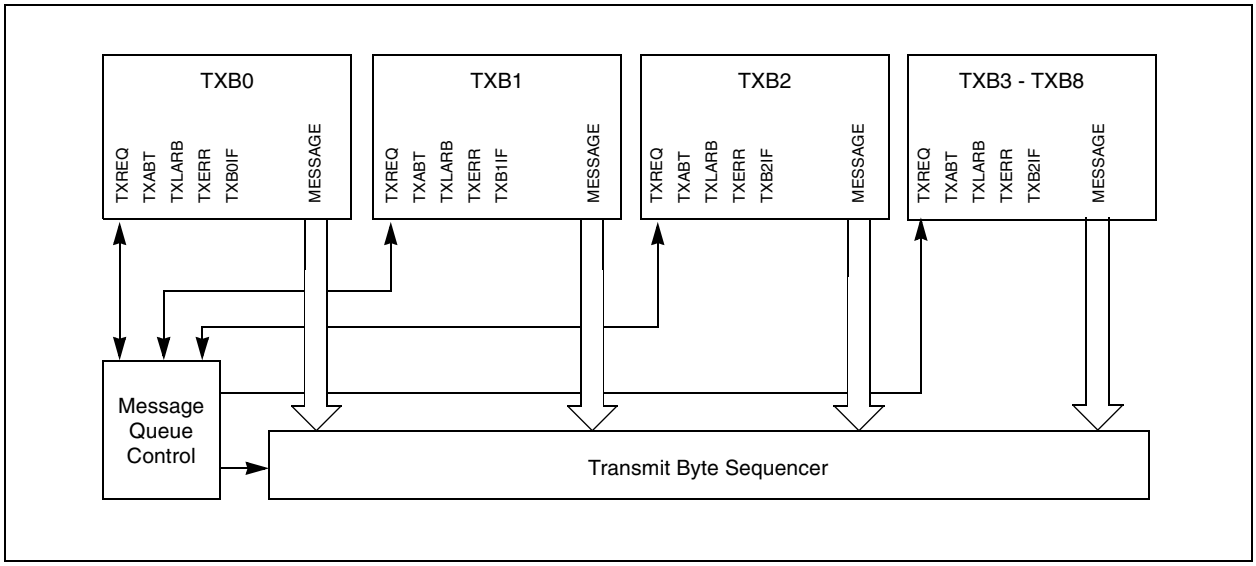


FIGURE 23-6: LENGTHENING A BIT PERIOD (ADDING SJW TO PHASE SEGMENT 1)

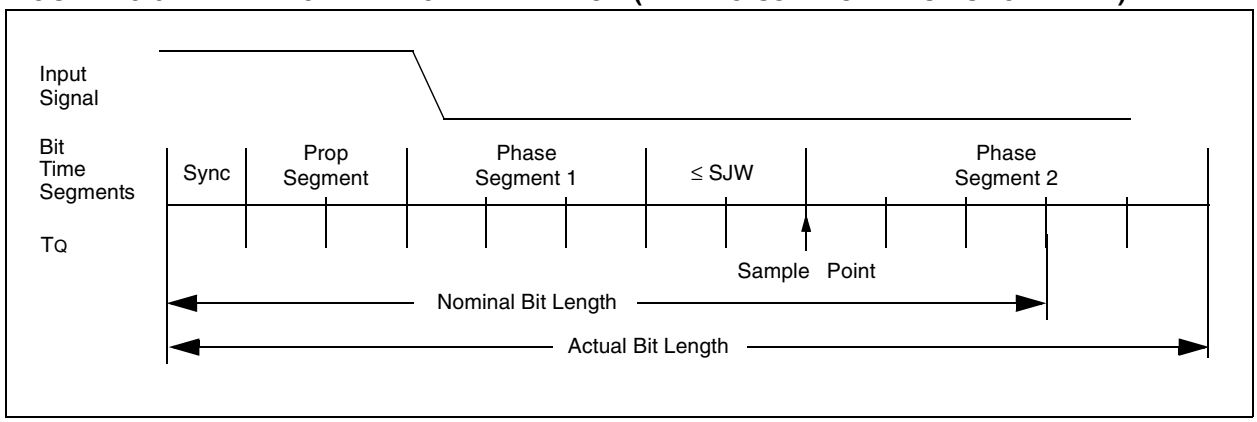
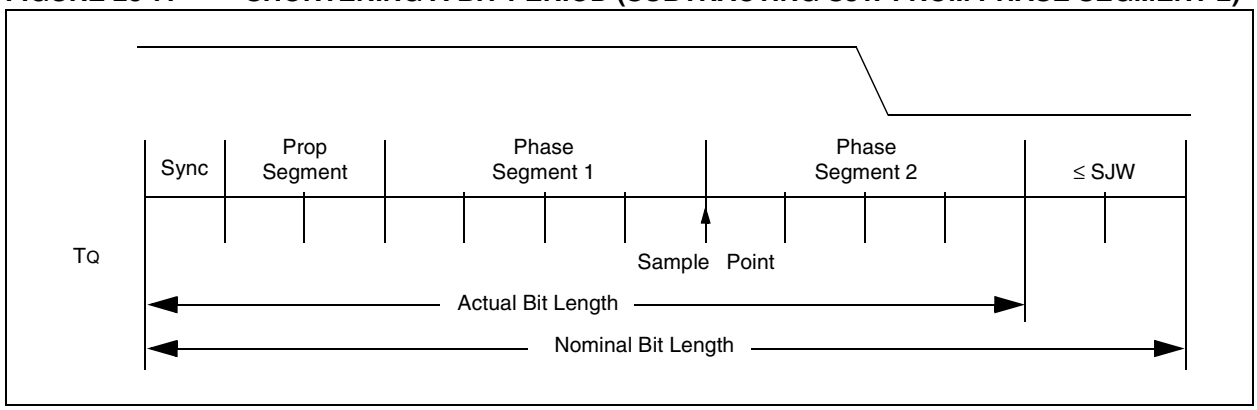


FIGURE 23-7: SHORTENING A BIT PERIOD (SUBTRACTING SJW FROM PHASE SEGMENT 2)



23.11 Programming Time Segments

Some requirements for programming of the time segments:

- Prop_Seg + Phase_Seg 1 ≥ Phase_Seg 2
- Phase_Seg 2 ≥ Sync Jump Width.

For example, assume that a 125 kHz CAN baud rate is desired, using 20 MHz for FOSC. With a TOSC of 50 ns, a baud rate prescaler value of 04h gives a TQ of 500 ns. To obtain a Nominal Bit Rate of 125 kHz, the Nominal Bit Time must be 8 µs or 16 TQ.

Using 1 TQ for the Sync_Seg, 2 TQ for the Prop_Seg and 7 TQ for Phase Segment 1 would place the sample point at 10 TQ after the transition. This leaves 6 TQ for Phase Segment 2.

By the rules above, the Sync Jump Width could be the maximum of 4 TQ. However, normally a large SJW is only necessary when the clock generation of the different nodes is inaccurate or unstable, such as using ceramic resonators. Typically, an SJW of 1 is enough.

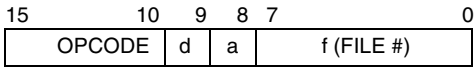
23.12 Oscillator Tolerance

As a rule of thumb, the bit timing requirements allow ceramic resonators to be used in applications with transmission rates of up to 125 Kbit/sec. For the full bus speed range of the CAN protocol, a quartz oscillator is required. A maximum node-to-node oscillator variation of 1.7% is allowed.

FIGURE 25-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

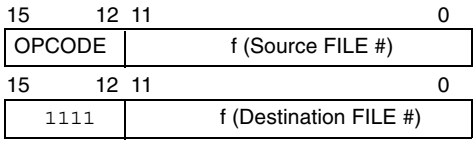
Example Instruction



ADDWF MYREG, W, B

d = 0 for result destination to be WREG register
d = 1 for result destination to be file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

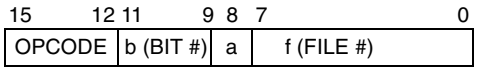
Byte to Byte move operations (2-word)



MOVFF MYREG1, MYREG2

f = 12-bit file register address

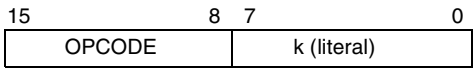
Bit-oriented file register operations



BSF MYREG, bit, B

b = 3-bit position of bit in file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

Literal operations

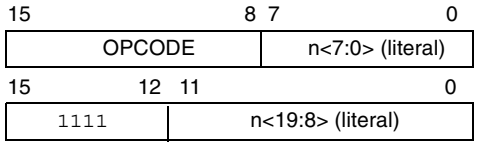


MOVLW 7Fh

k = 8-bit immediate value

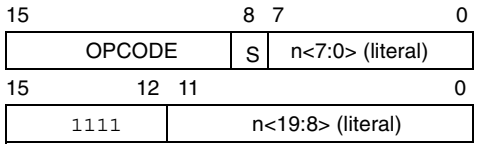
Control operations

CALL, GOTO and Branch operations



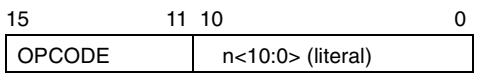
GOTO Label

n = 20-bit immediate value

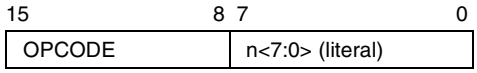


CALL MYFUNC

S = Fast bit



BRA MYFUNC



BC MYFUNC

MOVLW **Move Literal to W**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow W$

Status Affected: None

Encoding:

0000	1110	kkkk	kkkk

Description: The eight-bit literal 'k' is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: MOVLW 5Ah

After Instruction
W = 5Ah

MOVWF **Move W to f**

Syntax: MOVWF f {,a}

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

Operation: $(W) \rightarrow f$

Status Affected: None

Encoding:

0110	111a	ffff	ffff

Description: Move data from W to register 'f'. Location 'f' can be anywhere in the 256-byte bank.

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 25.2.3 "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: MOVWF REG, 0

Before Instruction
W = 4Fh
REG = FFh
After Instruction
W = 4Fh
REG = 4Fh

TBLRD Table Read

Syntax: TBLRD (*; *+; *-; +*)

Operands: None

Operation: if TBLRD *,
(Prog Mem (TBLPTR)) → TABLAT;
TBLPTR – No Change;
if TBLRD *+,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) + 1 → TBLPTR;
if TBLRD *-,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) – 1 → TBLPTR;
if TBLRD +*,
(TBLPTR) + 1 → TBLPTR;
(Prog Mem (TBLPTR)) → TABLAT;

Status Affected: None

Encoding:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer, called Table Pointer (TBLPTR), is used.

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word
TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

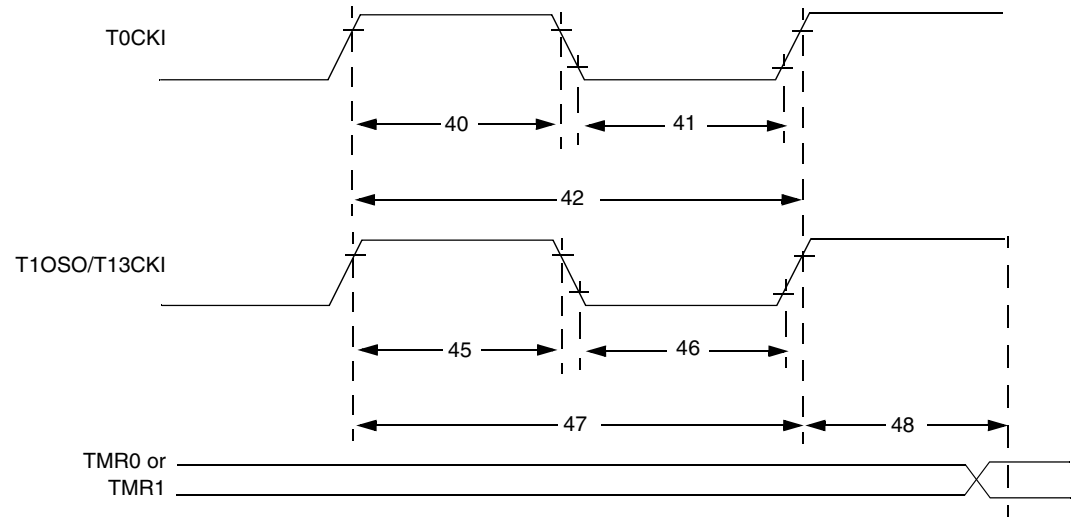
Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

TBLRD Table Read (Continued)

Example 1:
```
TBLRD   *+ ;
```
Before Instruction
TABLAT = 55h
TBLPTR = 00A356h
MEMORY(00A356h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 00A357h

Example 2:
```
TBLRD   +* ;
```
Before Instruction
TABLAT = 0AAh
TBLPTR = 01A357h
MEMORY(01A357h) = 12h
MEMORY(01A358h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 01A358h

FIGURE 27-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Note: Refer to Figure 27-4 for load conditions.

TABLE 27-11: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic			Min	Max	Units	Conditions
40	TT0H	T0CKI High Pulse Width		No prescaler	0.5 TCY + 20	—	ns	
				With prescaler	10	—	ns	
41	TT0L	T0CKI Low Pulse Width		No prescaler	0.5 TCY + 20	—	ns	
				With prescaler	10	—	ns	
42	TT0P	T0CKI Period		No prescaler	TCY + 10	—	ns	
				With prescaler	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4,..., 256)
45	TT1H	T13CKI High Time	Synchronous, no prescaler		0.5 TCY + 20	—	ns	
			Synchronous, with prescaler	PIC18**F**XXXX	10	—	ns	
				PIC18**LF**XXXX	25	—	ns	VDD = 2.0V
			Asynchronous	PIC18**F**XXXX	30	—	ns	
				PIC18**LF**XXXX	50	—	ns	VDD = 2.0V
46	TT1L	T13CKI Low Time	Synchronous, no prescaler		0.5 TCY + 5	—	ns	
			Synchronous, with prescaler	PIC18**F**XXXX	10	—	ns	
				PIC18**LF**XXXX	25	—	ns	VDD = 2.0V
			Asynchronous	PIC18**F**XXXX	30	—	ns	
				PIC18**LF**XXXX	50	—	ns	VDD = 2.0V
47	TT1P	T13CKI Input Period	Synchronous		Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous		60	—	ns	
	FT1	T13CKI Oscillator Input Frequency Range			DC	50	kHz	
48	TCKE2TMRI	Delay from External T13CKI Clock Edge to Timer Increment			2 TOSC	7 TOSC	—	

V

W

X