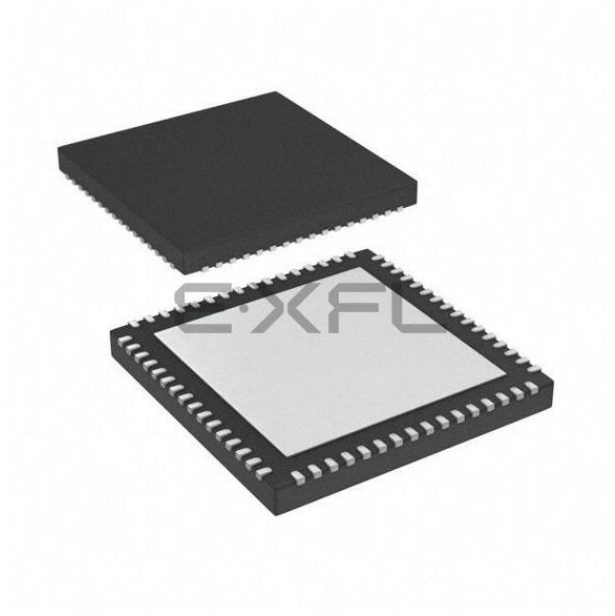


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66j94t-i-mr

TABLE 1-4: PIC18FXXJ94 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	100	80	64			
SEG19/AN0/AN1-/RP0/RA0	37	30	24			
SEG19				O	Analog	SEG19 output for LCD.
AN0				I	Analog	Analog Input 0.
AN1-				I	Analog	A/D negative input channel.
RP0				I/O	ST/DIG	Remappable Peripheral Pin 0 input/output.
RA0				I/O	ST/DIG	General purpose I/O pin.
SEG18/AN1/RP1/RA1	36	29	23			
SEG18				O	Analog	SEG18 output for LCD.
AN1				I	Analog	Analog Input 1.
RP1				I/O	ST/DIG	Remappable Peripheral Pin 1 input/output.
RA1				I/O	ST/DIG	General purpose I/O pin.
SEG21/VREF-/AN2/RP2/RA2	34	28	22			
SEG21				O	Analog	SEG21 output for LCD.
VREF-				I	Analog	A/D reference voltage (low) input.
AN2				I	Analog	Analog Input 2.
RP2				I/O	ST/DIG	Remappable Peripheral Pin 2 input/output.
RA2				I/O	ST/DIG	General purpose I/O pin.
VREF+/AN3/RP3/RA3	33	27	21			
VREF+				I	Analog	A/D reference voltage (high) input.
AN3				I	Analog	Analog Input 3.
RP3				I/O	ST/DIG	Remappable Peripheral Pin 3 input/output.
RA3				I/O	ST/DIG	General purpose I/O pin.
SEG14/AN6/RP4/RA4	43	34	28			
SEG14				O	Analog	SEG14 output for LCD.
AN6				I	Analog	Analog Input 6.
RP4				I/O	ST/DIG	Remappable Peripheral Pin 4 input/output.
RA4				I/O	ST/DIG	General purpose I/O pin.
SEG15/AN4/LVDIN/C1INA/ C2INA/C3INA/RP5/RA5	42	33	27			
SEG15				O	Analog	SEG15 output for LCD.
AN4				I	Analog	Analog Input 4.
LVDIN				I	Analog	High/Low-Voltage Detect (HLVD) input.
C1INA				I	Analog	Comparator 1 Input A.
C2INA				I	Analog	Comparator 2 Input A.
C3INA				I	Analog	Comparator 3 Input A.
RP5				I/O	ST/DIG	Remappable Peripheral Pin 5 input/output.
RA5				I/O	ST/DIG	General purpose I/O pin.

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
I^2C = I^2C/SMBus
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output
OD = Open-Drain (no P diode to VDD)

TABLE 1-4: PIC18FXXJ94 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	100	80	64			
SOSCO/SCLKI/PWRLCLK/RC0	45	36	30			
SOSCO				O	—	SOSC oscillator output.
SCLKI				I	ST	Digital SOSC input.
PWRLCLK				I	ST	SOSC input at 50 Hz or 60 Hz only (RTCCLKSEL<1:0> = 11 or 10).
RC0				I/O	ST	General purpose Input pin.
SOSCI/RC1	44	35	29			
SOSCI				I	Analog	Timer1 oscillator input.
RC1				I/O	ST	General purpose Input pin.
SEG13/AN9/RP11/CTED7/RC2	53	43	33			
SEG13				O	Analog	SEG13 output for LCD.
AN9				I	Analog	Analog Input 9.
RP11				I/O	ST/DIG	Remappable Peripheral Pin 11 input/output.
CTED7				I	ST	CTMU Edge 7 input.
RC2				I/O	ST/DIG	General purpose I/O pin.
SEG17/SCL1/RP15/CTED8/RC3	54	44	34			
SEG17				O	Analog	SEG17 output for LCD.
SCL1				I/O	I^2C	I^2C clock input/output.
RP15				I/O	ST/DIG	Remappable Peripheral Pin 15 input/output.
CTED8				I	ST	CTMU Edge 8 input.
RC3				I/O	ST/DIG	General purpose I/O pin.
SEG16/SDA1/RP17/CTED9/RC4	56	45	35			
SEG16				O	Analog	SEG16 output for LCD.
SDA1				I/O	I^2C	I^2C data input/output.
RP17				I/O	ST/DIG	Remappable Peripheral Pin 17 input/output.
CTED9				I	ST	CTMU Edge 9 input.
RC4				I/O	ST/DIG	General purpose I/O pin.
SEG12/RP16/CTED10/RC5	57	46	36			
SEG12				O	Analog	SEG12 output for LCD.
RP16				I/O	ST/DIG	Remappable Peripheral Pin 16 input/output.
CTED10				I	ST	CTMU Edge 10 input.
RC5				I/O	ST/DIG	General purpose I/O pin.
SEG27/RP18/$\overline{UOE}$/CTED11/RC6	47	37	31			
SEG27				O	Analog	SEG27 output for LCD.
RP18				I/O	ST/DIG	Remappable Peripheral Pin 18 input/output.
$\overline{UOE}$/				O	DIG	External USB transceiver NOE output.
CTED11				I	ST	CTMU Edge 11 input.
RC6				I/O	ST/DIG	General purpose I/O pin.
SEG22/RP19/CTED12/RC7	48	38	32			
SEG22				O	Analog	SEG22 output for LCD.
RP19				I/O	ST/DIG	Remappable Peripheral Pin 19 input/output.
CTED12				I	ST	CTMU Edge 12 input.
RC7				I/O	ST/DIG	General purpose I/O pin.

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

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS, is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 μF (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 μF in parallel with 0.001 μF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 μF to 47 μF.

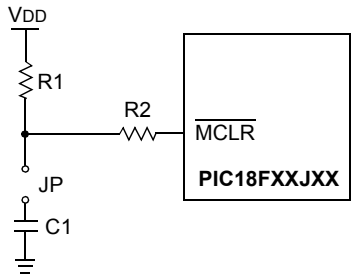
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides two specific device functions: Device Reset, and Device Programming and Debugging. If programming and debugging are not required in the end application, a direct connection to VDD may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF $\overline{\text{MCLR}}$ PIN CONNECTIONS



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into $\overline{\text{MCLR}}$ from the external capacitor, C, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

3.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSC1 and OSC2 pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins
- Fast Internal RC (FRC) Oscillator
- Low-Power Internal RC (LPRC) Oscillator

The Primary Oscillator and FRC sources have the option of using the internal USB PLL block, which generates both the USB module clock and a separate system clock from the 96 MHz PLL. Refer to **Section 3.8.1 "Oscillator Modes and USB Operation"** for additional information.

The internal FRC provides an 8 MHz clock source. It can optionally be reduced by the programmable clock divider to provide a range of system clock frequencies.

The selected clock source generates the processor and peripheral clock sources. The processor clock source is divided by four to produce the internal instruction cycle clock, FCY. In this document, the instruction cycle clock is also denoted by FOSC/4. The internal instruction cycle clock, FOSC/4, can be provided on the OSC2 I/O pin for some operating modes of the Primary Oscillator. The timing diagram in Figure 3-2 shows the relationship between the processor clock source and instruction execution.

FIGURE 3-2: CLOCK OR INSTRUCTION CYCLE TIMING

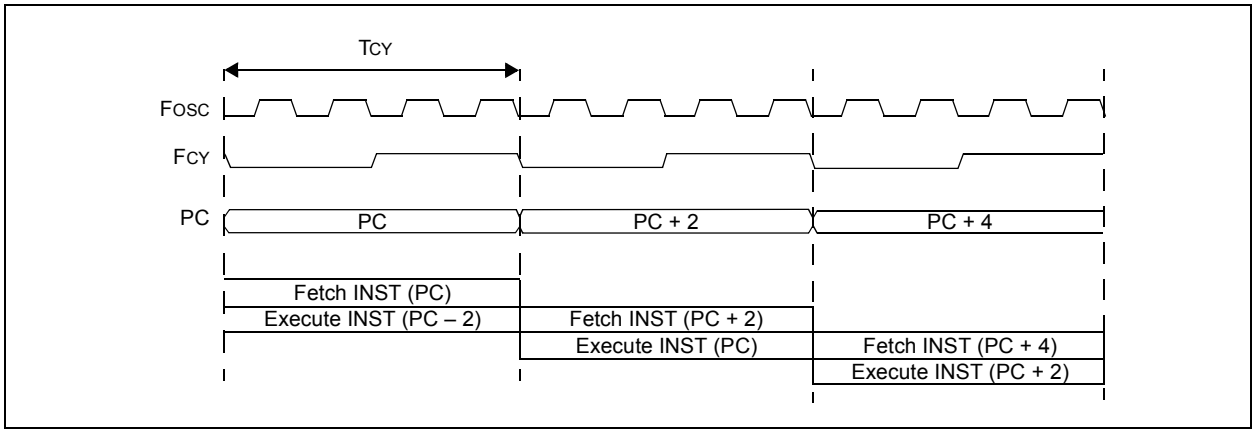


TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

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
RCON4	64-pin	80-pin	100-pin	00-0 -0-0	00-u -0-u	00-u -0-u
UFRML	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
UFRMH	64-pin	80-pin	100-pin	---- -xxx	---- -xxx	---- -uuu
UIR	64-pin	80-pin	100-pin	-000 0000	-000 0000	-uuu uuuu
UEIR	64-pin	80-pin	100-pin	-000 0000	-000 0000	-uuu uuuu
USTAT	64-pin	80-pin	100-pin	0--0 0000	0--0 0000	u--u uuuu
UCON	64-pin	80-pin	100-pin	-0x0 000-	-0x0 000-	-uuu uuu-
UADDR	64-pin	80-pin	100-pin	-000 0000	-000 0000	-uuu uuuu
TRISVP	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
LATVP	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTVP	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
TXADDRL	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
TXADDRH	64-pin	80-pin	100-pin	---- 0000	---- 0000	---- uuuu
RXADDRL	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RXADDRH	64-pin	80-pin	100-pin	---- 0000	---- 0000	---- uuuu
DMABCL	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
DMABCH	64-pin	80-pin	100-pin	---- --00	---- --00	---- --uu
TXBUF	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP1CON3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP1MSK	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
BAUDCON1	64-pin	80-pin	100-pin	0100 0000	0100 0000	uuuu uuuu
OSCCON2	64-pin	80-pin	100-pin	000- 000-	00q- 000-	uuu- uuu-
OSCCON3	64-pin	80-pin	100-pin	---- -001	---- -uuu	---- -uuu
OSCCON4	64-pin	80-pin	100-pin	000- ----	uuu- ----	uuu- ----
OSCCON5	64-pin	80-pin	100-pin	0-00 0000	u-uu uuuu	u-uu uuuu
WPUB	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
PIE6	64-pin	80-pin	100-pin	0000 -000	0000 -000	uuuu -uuu
DMACON1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RTCCON1	64-pin	80-pin	100-pin	0-00 0000	u-uu uuuu	u-uu uuuu
RTCCAL	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
RTCVALH	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
RTCVALL	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
ALRMCFG	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu

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

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 5-2 for Reset value for specific condition.

5: Bits 7,6 are unimplemented on 64 and 80-pin devices.

6: If the V_{BAT} is always powered, the DSGPx register values will remain unchanged after the first POR.

6.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value.

These operands are:

- POSTDEC – Accesses the FSR value, then automatically decrements it by ‘1’ afterwards
- POSTINC – Accesses the FSR value, then automatically increments it by ‘1’ afterwards
- PREINC – Increments the FSR value by ‘1’, then uses it in the operation
- PLUSW – Adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value, offset by the value in the W register, with neither value actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair. Rollovers of the FSRnL register, from FFh to 00h, carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (for example, Z, N and OV bits).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

6.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations.

As a specific case, assume that the FSR0H:FSR0L registers contain FE7h, the address of INDF1. Attempts to read the value of the INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a NOP.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair, but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, however, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution, so that they do not inadvertently change settings that might affect the operation of the device.

6.5 Program Memory and the Extended Instruction Set

The operation of program memory is unaffected by the use of the extended instruction set.

Enabling the extended instruction set adds five additional two-word commands to the existing PIC18 instruction set: ADDFSR, CALLW, MOVSF, MOVSS and SUBFSR. These instructions are executed as described in **Section 6.2.4 “Two-Word Instructions”**.

6.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

The use of Indexed Literal Offset Addressing mode effectively changes how the lower part of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom part of Bank 0, this mode maps the contents from Bank 0 and a user-defined "window" that can be located anywhere in the data memory space.

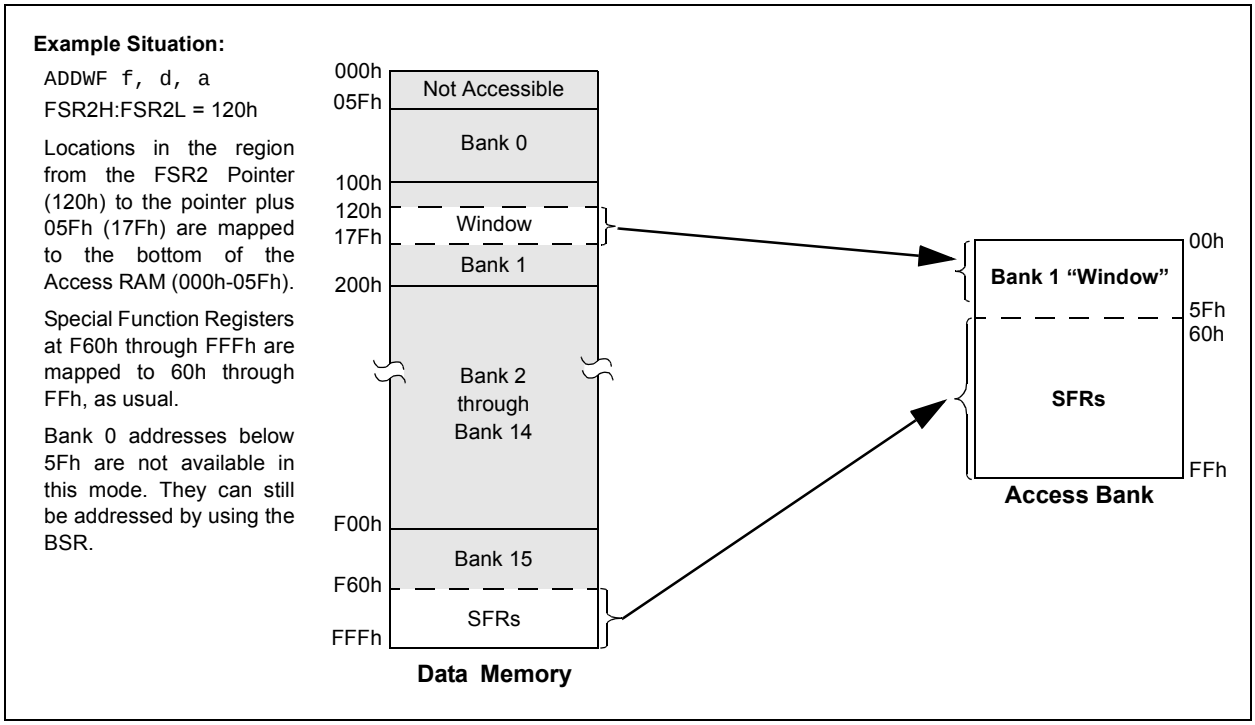
The value of FSR2 establishes the lower boundary of the addresses mapped into the window, while the upper boundary is defined by FSR2 plus 95 (5Fh). Addresses in the Access RAM above 5Fh are mapped as previously described. (See **Section 6.3.2 "Access Bank"**.) An example of Access Bank remapping in this addressing mode is shown in Figure 6-10.

Remapping the Access Bank applies *only* to operations using the Indexed Literal Offset mode. Operations that use the BSR (Access RAM bit = 1) will continue to use Direct Addressing as before. Any Indirect or Indexed Addressing operation that explicitly uses any of the indirect file operands (including FSR2) will continue to operate as standard Indirect Addressing. Any instruction that uses the Access Bank, but includes a register address of greater than 05Fh, will use Direct Addressing and the normal Access Bank map.

6.6.4 BSR IN INDEXED LITERAL OFFSET MODE

Although the Access Bank is remapped when the extended instruction set is enabled, the operation of the BSR remains unchanged. Direct Addressing, using the BSR to select the data memory bank, operates in the same manner as previously described.

FIGURE 6-10: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING



8.8 Operation in Power-Managed Modes

In alternate, power-managed Run modes, the external bus continues to operate normally. If a clock source with a lower speed is selected, bus operations will run at that speed. In these cases, excessive access times for the external memory may result if Wait states have been enabled and added to external memory operations. If operations in a lower power Run mode are anticipated, users should provide in their applications for adjusting memory access times at the lower clock speeds.

In Sleep and Idle modes, the microcontroller core does not need to access data; bus operations are suspended. The state of the external bus is frozen, with the address/data pins and most of the control pins holding at the same state they were in when the mode was invoked. The only potential changes are to the $\overline{CE}$, $\overline{LB}$ and $\overline{UB}$ pins, which are held at logic high.

TABLE 8-3: REGISTERS ASSOCIATED WITH THE EXTERNAL MEMORY BUS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MEMCON(1)	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
PADCFG	RDPU	REPU	RFPU	RGPU	RHPU	RJPU	RKPU	RLPU
PMD4	CMP1MD	CMP2MD	CMP3MD	USBMD	IOCMD	LVDMD	—	EMBMD

Legend: — = unimplemented, read as '0'. Shaded cells are not used during External Memory Bus access.

Note 1: This register is unimplemented on 64-pin devices read as '0'.

FIGURE 12-2: ON-OFF KEYING (OOK) SYNCHRONIZATION

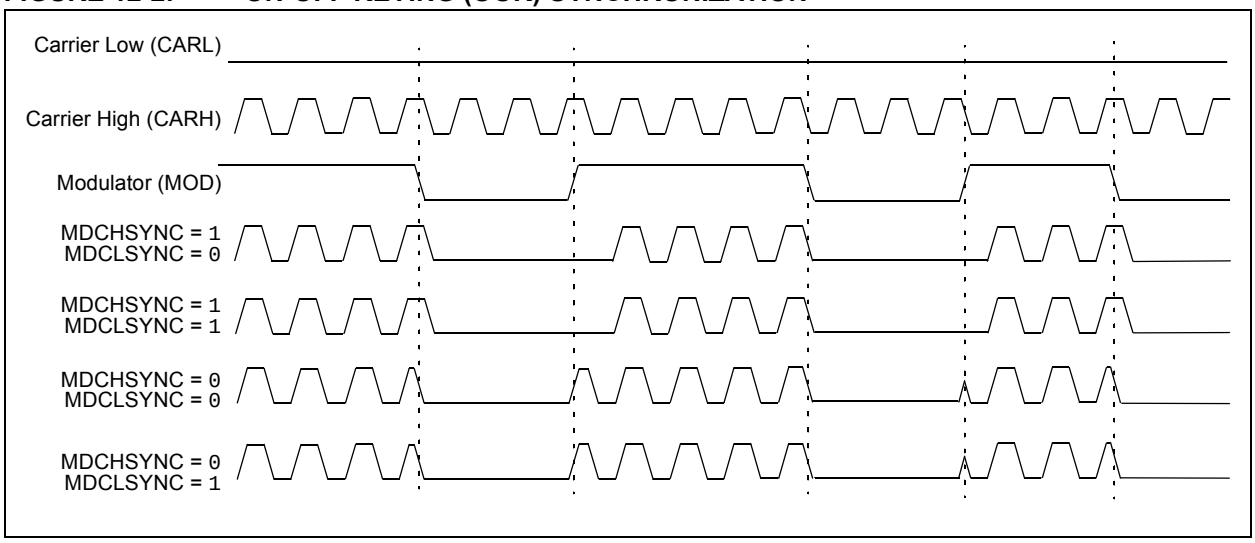


FIGURE 12-3: NO SYNCHRONIZATION (MDCHSYNC = 0, MDCLSYNC = 0)

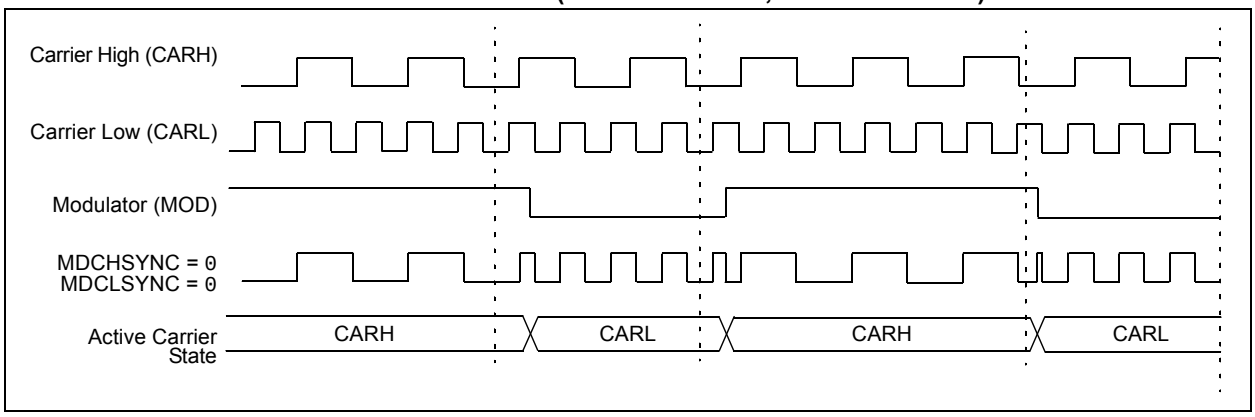
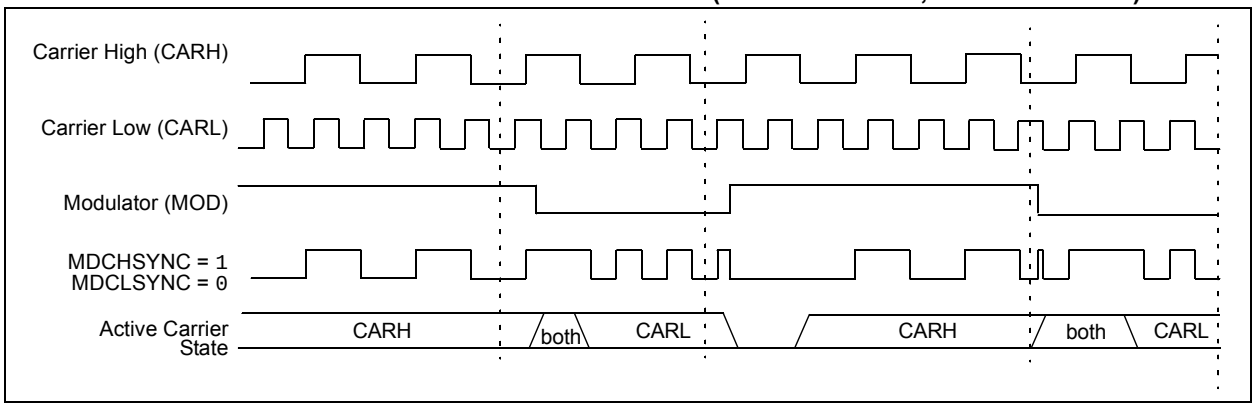


FIGURE 12-4: CARRIER HIGH SYNCHRONIZATION (MDCHSYNC = 1, MDCLSYNC = 0)



REGISTER 13-2: LCDREG: LCD CHARGE PUMP CONTROL REGISTER

R/W-0	U-0	RW-1	RW-1	RW-1	RW-1	RW-0	RW-0
CPEN	—	BIAS2	BIAS1	BIAS0	MODE13	CLKSEL1	CLKSEL0
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 **CPEN:** 3.6V Charge Pump Enable bit
1 = The regulator generates the highest (3.6V) voltage
0 = Highest voltage in the system is supplied externally (V_{DD})

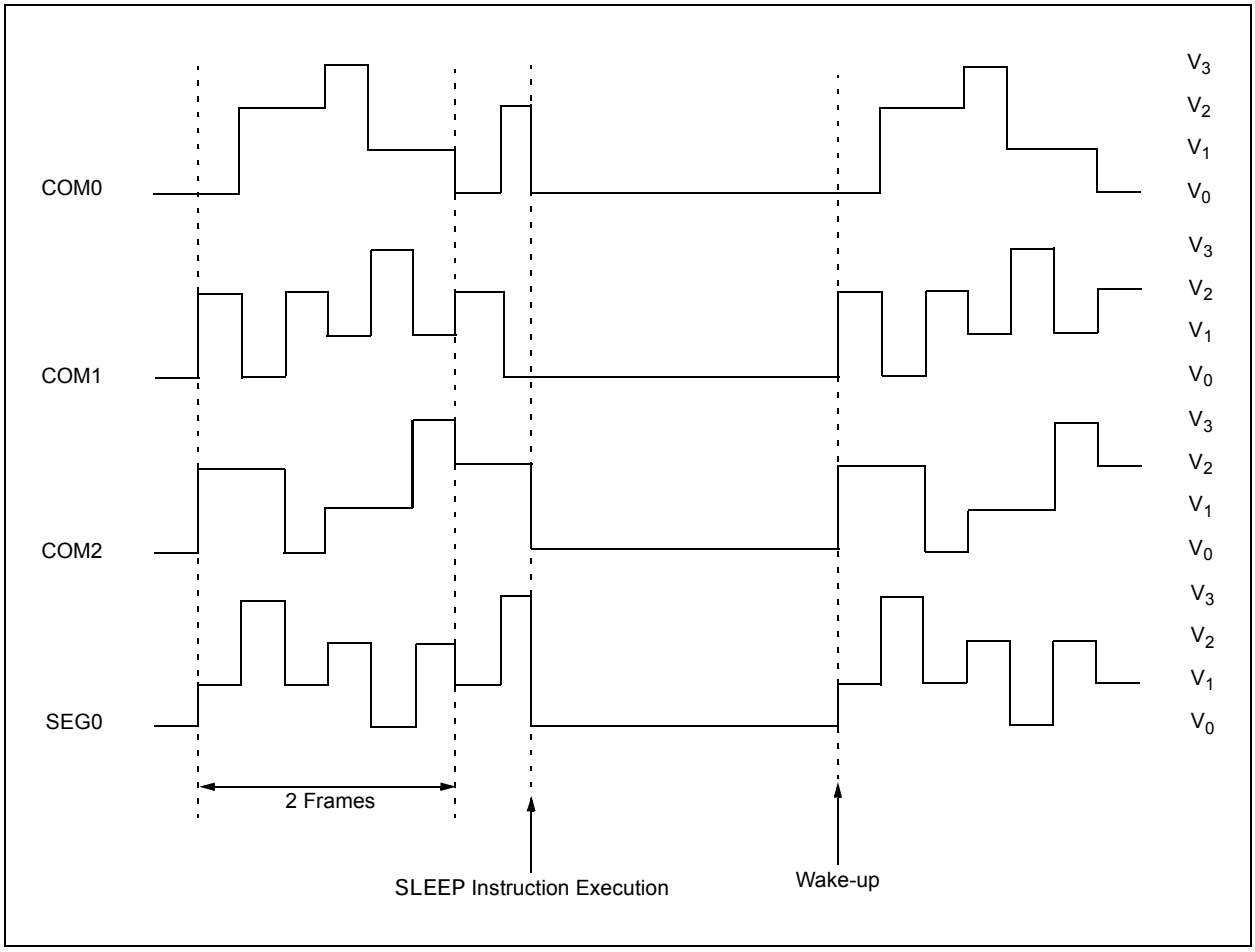
bit 6 **Unimplemented:** Read as '0'

bit 5-3 **BIAS<2:0>:** Regulator Voltage Output Control bits
111 =3.60V peak (offset on LCDBIAS0 of 0V)
110 =3.47V peak (offset on LCDBIAS0 of 0.13V)
101 =3.34V peak (offset on LCDBIAS0 of 0.26V)
100 =3.21V peak (offset on LCDBIAS0 of 0.39V)
011 =3.08V peak (offset on LCDBIAS0 of 0.52V)
010 =2.95V peak (offset on LCDBIAS0 of 0.65V)
001 =2.82V peak (offset on LCDBIAS0 of 0.78V)
000 =2.69V peak (offset on LCDBIAS0 of 0.91V)

bit 2 **MODE13:** 1/3 LCD BIAS Enable bit
1 = Regulator output supports 1/3 LCD BIAS mode
0 = Regulator output supports Static LCD BIAS mode

bit 1-0 **CLKSEL<1:0>**: Regulator Clock Select Control bits
11 =LPRC
10 =FRC
01 =SOSC
00 =Disable regulator and float regulator voltage output.

FIGURE 13-23: SLEEP ENTRY/EXIT WHEN SLPEN = 1 OR CS<1:0> = 00.



When a shutdown event occurs, two things happen:

- The ECCPxASE bit is set to '1'. The ECCPxASE will remain set until cleared in firmware or an auto-restart occurs. (See **Section 18.4.5 "Auto-Restart Mode"**.)
- The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs (PxA/PxC and PxB/PxD). The state of each pin pair is determined by the PSSxAC and PSSxBD bits (ECCPxAS<3:2> and <1:0>, respectively).

Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

REGISTER 18-3: ECCPxAS: ECCPx AUTO-SHUTDOWN CONTROL REGISTER(1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0
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 **ECCPxASE:** ECCP Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in a shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPxAS<2:0>:** ECCP Auto-Shutdown Source Select bits
000 = Auto-shutdown is disabled
001 = Comparator C1OUT output is high
010 = Comparator C2OUT output is high
011 = Either Comparator C1OUT or C2OUT is high
100 = VIL on FLT0 pin
101 = VIL on FLT0 pin or Comparator C1OUT output is high
110 = VIL on FLT0 pin or Comparator C2OUT output is high
111 = VIL on FLT0 pin or Comparator C1OUT or Comparator C2OUT is high

bit 3-2 **PSSxAC<1:0>:** PxA and PxC Pins Shutdown State Control bits
00 = Drive pins: PxA and PxC to '0'
01 = Drive pins: PxA and PxC to '1'
1x = PxA and PxC pins tri-state

bit 1-0 **PSSxBD<1:0>:** Pins PxB and PxD Shutdown State Control bits
00 = Drive pins: PxB and PxD to '0'
01 = Drive pins: PxB and PxD to '1'
1x = PxB and PxD pins tri-state

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
2: Writing to the ECCPxASE bit is disabled while an auto-shutdown condition persists.
3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

REGISTER 20-5: DMACON2: DMA CONTROL REGISTER 2 (CONTINUED)

bit 3-0 **INTLVL<3:0>:** Watermark Interrupt Enable bits
These bits specify the amount of remaining data yet to be transferred (transmitted and/or received) upon which an interrupt is generated.

1111 = Amount of remaining data to be transferred is 576 bytes
1110 = Amount of remaining data to be transferred is 512 bytes
1101 = Amount of remaining data to be transferred is 448 bytes
1100 = Amount of remaining data to be transferred is 384 bytes
1011 = Amount of remaining data to be transferred is 320 bytes
1010 = Amount of remaining data to be transferred is 256 bytes
1001 = Amount of remaining data to be transferred is 192 bytes
1000 = Amount of remaining data to be transferred is 128 bytes
0111 = Amount of remaining data to be transferred is 67 bytes
0110 = Amount of remaining data to be transferred is 32 bytes
0101 = Amount of remaining data to be transferred is 16 bytes
0100 = Amount of remaining data to be transferred is 8 bytes
0011 = Amount of remaining data to be transferred is 4 bytes
0010 = Amount of remaining data to be transferred is 2 bytes
0001 = Amount of remaining data to be transferred is 1 byte
0000 = Transfer complete

20.4.4.3 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 20-1.

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 20-1: BYTES TRANSMITTED FOR A GIVEN DMABC

$$Bytes_{XMIT} = (DMABC + 1)$$

20.4.4.4 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, including memory used for USB endpoint buffers. The SPI DMA module cannot be used to read from the Special Function Registers (SFRs) contained in Banks 14 and 15.

20.4.4.5 RXADDRH and RXADDRL

The RXADDRH and RXADDRL registers 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.

REGISTER 26-2: CTMUCON2: CTMU CURRENT CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
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 7-2 **ITRIM<5:0>:** Current Source Trim bits

011111 = Maximum positive change (+62% typ.) from nominal current
011110
.
.
.
000001 = Minimum positive change (+2% typ.) from nominal current
000000 = Nominal current output specified by IRNG<1:0>
111111 = Minimum negative change (-2% typ.) from nominal current
.
.
.
100010
100001 = Maximum negative change (-62% typ.) from nominal current

bit 1-0 **IRNG<1:0>:** Current Source Range Select bits

11 = 100 x Base Current
10 = 10 x Base Current
01 = Base Current Level (0.55 μA nominal)
00 = 1000 x Base Current

EXAMPLE 26-2: CTMU CURRENT CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 500                       //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027                       //R value is 4200000 (4.2M)
                                        //scaled so that result is in
                                        //1/100th of uA
#define ADSCALE 1023                    //for unsigned conversion 10 sig bits
#define ADREF 3.3                       //Vdd connected to A/D Vr+

int main(void)
{
    int i;
    int j = 0; //index for loop
    unsigned int Vread = 0;
    double VTot = 0;
    float Vavg=0, Vcal=0, CTMUISrc = 0;  //float values stored for calcs

//assume CTMU and A/D have been setup correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONbits.CTMUEN = 1;                 //Enable the CTMU
    for(j=0;j<10;j++)
    {
        CTMUCONbits.IDISSEN = 1;        //drain charge on the circuit
        DELAY;                          //wait 125us
        CTMUCONbits.IDISSEN = 0;        //end drain of circuit

        CTMUCON3bits.EDG1STAT = 1;      //Begin charging the circuit
                                        //using CTMU current source
        DELAY;                          //wait for 125us
        CTMUCON3bits.EDG1STAT = 0;      //Stop charging circuit

        PIR1bits.ADIF = 0;              //make sure A/D Int not set
        ADCON1Lbits.SAMP=1;             //and begin A/D conv.
        while(!PIR1bits.ADIF);          //Wait for A/D convert complete

        Vread = ADRES;                  //Get the value from the A/D
        PIR1bits.ADIF = 0;              //Clear A/D Interrupt Flag
        VTot += Vread;                  //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);        //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;               //CTMUISrc is in 1/100ths of uA

}
```

EXAMPLE 26-3: CTMU CAPACITANCE CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 25                        //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5                 //time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023                    //for unsigned conversion 10 sig bits
#define ADREF 3.3                       //Vdd connected to A/D Vr+
#define RCAL .027                       //R value is 4200000 (4.2M)
                                        //scaled so that result is in
                                        //1/100th of uA

int main(void)
{
    int i;
    int j = 0;                          //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

//assume CTMU and A/D have been setup correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONbits.CTMUEN = 1;                 //Enable the CTMU
    for(j=0;j<10;j++)
    {
        CTMUCONbits.IDISSEN = 1;        //drain charge on the circuit
        DELAY;                          //wait 125us
        CTMUCONbits.IDISSEN = 0;        //end drain of circuit

        CTMUCON3bits.EDG1STAT = 1;      //Begin charging the circuit
                                        //using CTMU current source
        DELAY;                          //wait for 125us
        CTMUCON3bits.EDG1STAT = 0;      //Stop charging circuit

        PIR1bits.ADIF = 0;              //make sure A/D Int not set
        ADCON1Lbits.SAMP=1;             //and begin A/D conv.
        while(!PIR1bits.ADIF);          //Wait for A/D convert complete


        Vread = ADRES;                  //Get the value from the A/D
        PIR1bits.ADIF = 0;              //Clear A/D Interrupt Flag
        VTot += Vread;                  //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);        //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;               //CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

RETFIE **Return from Interrupt**

Syntax: RETFIE {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
1 → GIE/GIEH or PEIE/GIEL;
if s = 1,
(WS) → W,
(STATUSS) → STATUS,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: GIE/GIEH, PEIE/GIEL.

Encoding:

0000	0000	0001	000s

Description: Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low-priority Global Interrupt Enable bit. If 's' = 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	POP PC from stack Set GIEH or GIEL
No operation	No operation	No operation	No operation

Example: RETFIE 1

After Interrupt
PC = TOS
W = WS
BSR = BSRS
STATUS = STATUSS
GIE/GIEH, PEIE/GIEL = 1

RETLW **Return Literal to W**

Syntax: RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W,
(TOS) → PC,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	1100	kkkk	kkkk

Description: W is loaded with the 8-bit literal 'k'. The Program Counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	POP PC from stack, write to W
No operation	No operation	No operation	No operation

Example:

```
    CALL TABLE ; W contains table
               ; offset value
               ; W now has
               ; table value
    :
TABLE
    ADDWF PCL  ; W = offset
    RETLW k0   ; Begin table
    RETLW k1   ;
    :
    :
    RETLW kn   ; End of table
```

Before Instruction
W = 07h
After Instruction
W = value of kn

30.1.2 TIMING CONDITIONS

The temperature and voltages specified in Table 30-18 apply to all timing specifications unless otherwise noted. Figure 30-2 specifies the load conditions for the timing specifications.

TABLE 30-18: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial Operating voltage VDD range as described in **Section TABLE 30-1:** and **Section** .

FIGURE 30-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

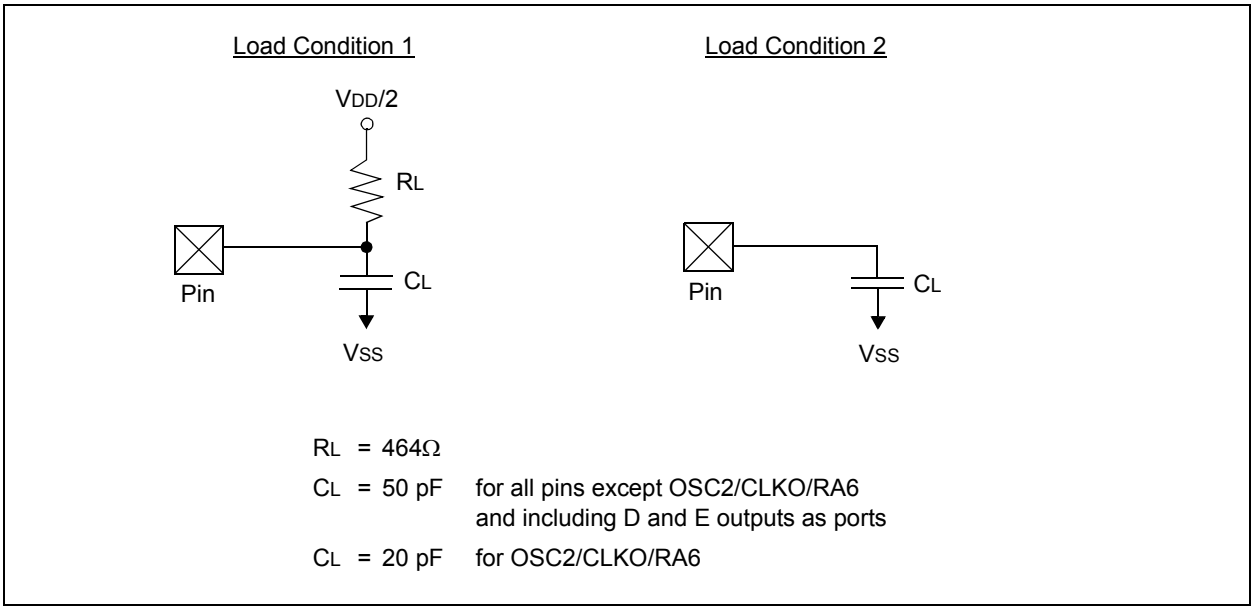
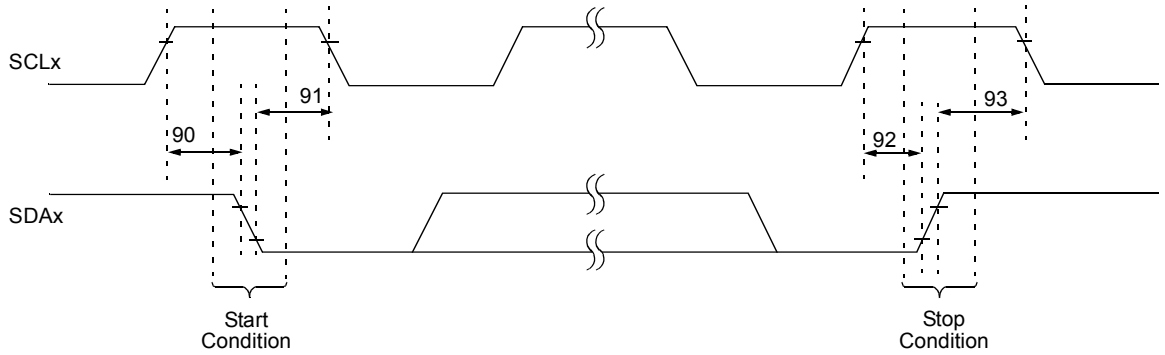


FIGURE 30-16: I²C BUS START/STOP BITS TIMING



Note: Refer to Figure 30-2 for load conditions.

TABLE 30-34: I²C BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4700	—	ns	
			400 kHz mode	600	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—		