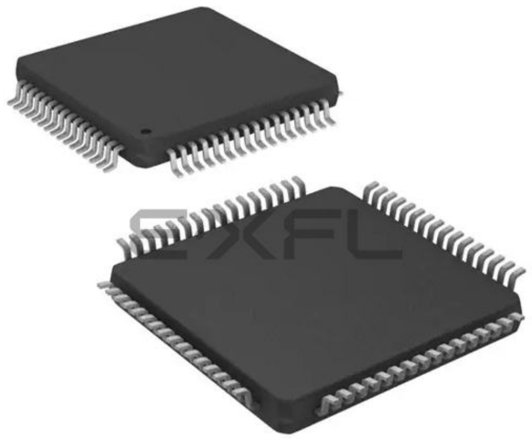


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

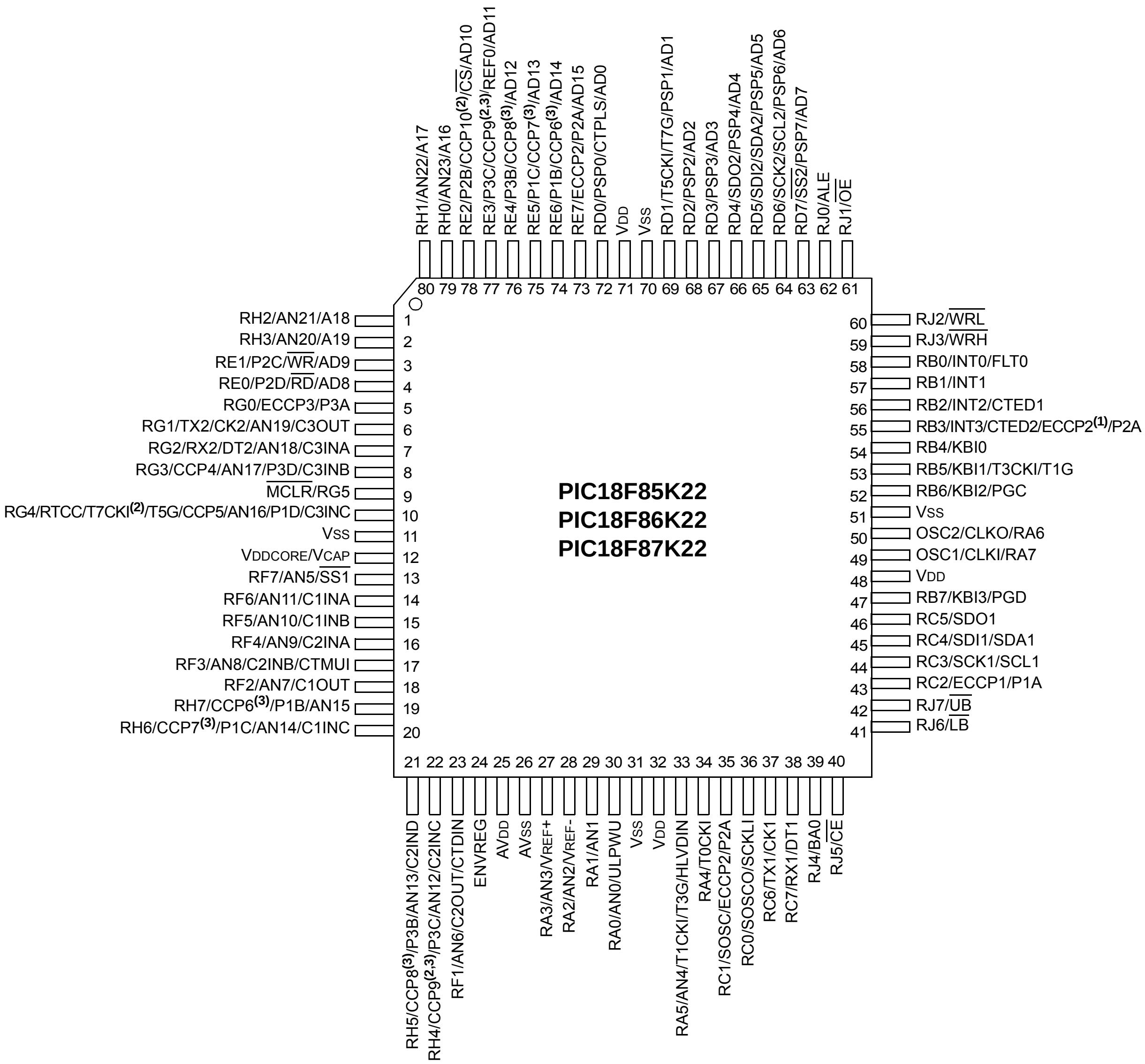
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f67k22t-i-pt

Pin Diagrams – PIC18F8XK22

80-Pin TQFP

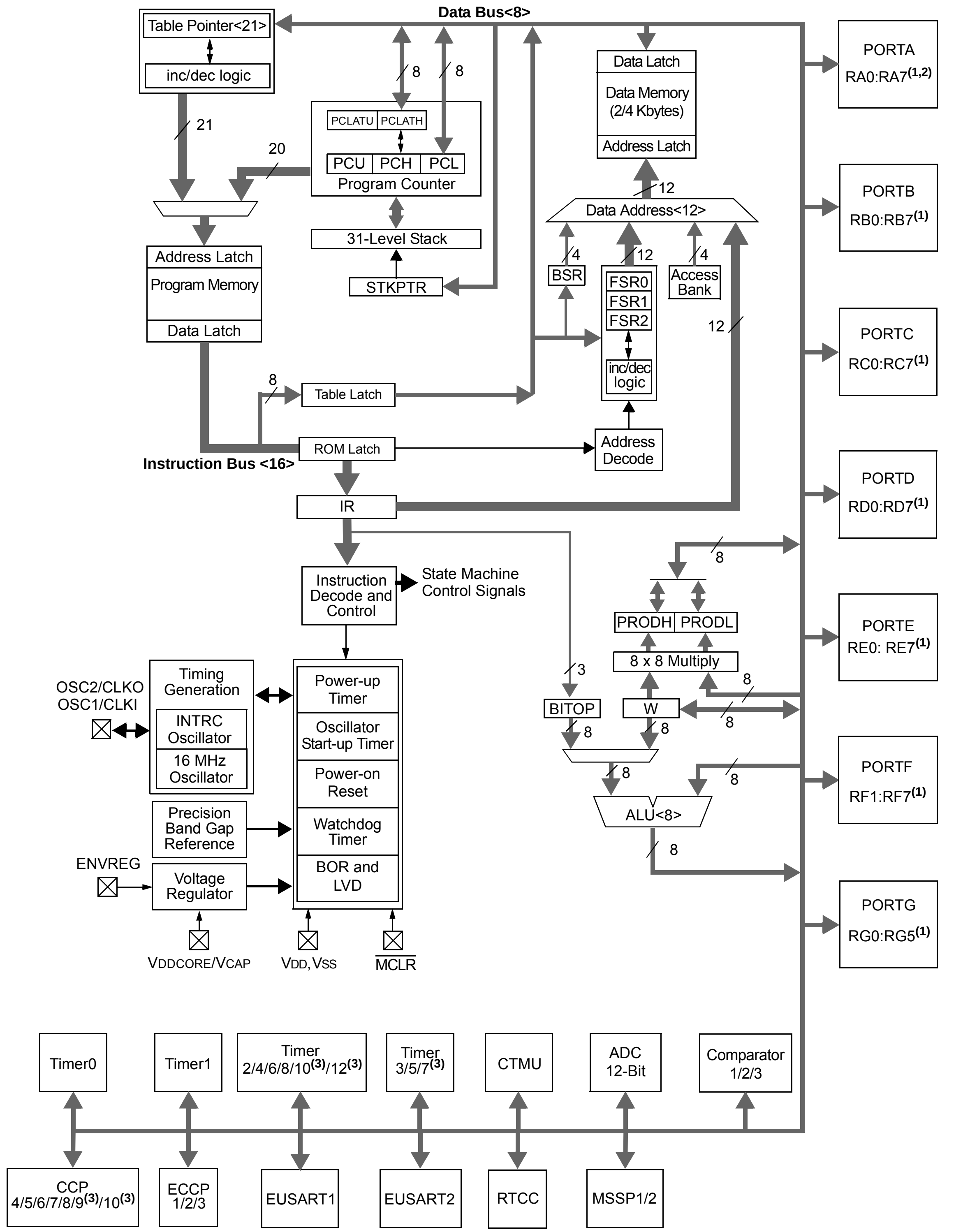


Note 1: The ECCP2 pin placement depends on the CCP2MX Configuration bit setting and whether the device is in Microcontroller or Extended Microcontroller mode.

2: Not available on the PIC18F65K22 and PIC18F85K22 devices.

3: The CC6, CCP7, CCP8 and CCP9 pin placement depends on the setting of the ECCPMX Configuration bit (CONFIG3H<1>).

FIGURE 1-1: PIC18F6XK22 (64-PIN) BLOCK DIAGRAM



Note 1: See Table 1-3 for I/O port pin descriptions.

2: RA6 and RA7 are only available as digital I/O in select oscillator modes. For more information, see **Section 3.0 "Oscillator Configurations"**.

3: Unimplemented on the PIC18F65K22.

REGISTER 3-3: OSCTUNE: OSCILLATOR TUNING REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
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 **INTSRC:** Internal Oscillator Low-Frequency Source Select bit
1 = 31.25 kHz device clock derived from 16 MHz INTOSC source (divide-by-512 enabled, HF-INTOSC)
0 = 31 kHz device clock derived from INTRC 31 kHz oscillator (LF-INTOSC)

bit 6 **PLLEN:** Frequency Multiplier PLL Enable bit
1 = PLL is enabled
0 = PLL is disabled

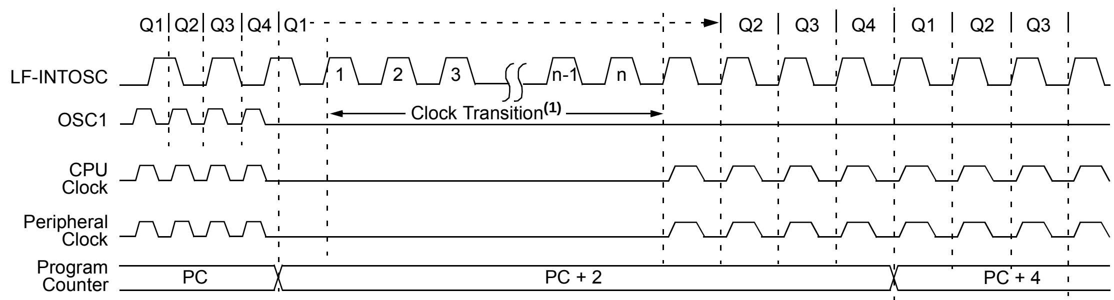
bit 5-0 **TUN<5:0>:** Fast RC Oscillator (INTOSC) Frequency Tuning bits
011111 = Maximum frequency
• •
• •
000001
000000 = Center frequency; fast RC oscillator is running at the calibrated frequency
111111
• •
• •
100000 = Minimum frequency

Clocks to the device continue while the INTOSC source stabilizes after an interval of TIOBST (Parameter 39, Table 31-13).

If the IRCF bits were previously at a non-zero value, or if INTSRC was set before setting SCS1 and the INTOSC source was already stable, the HFIOFS or MFIOFS bit will remain set.

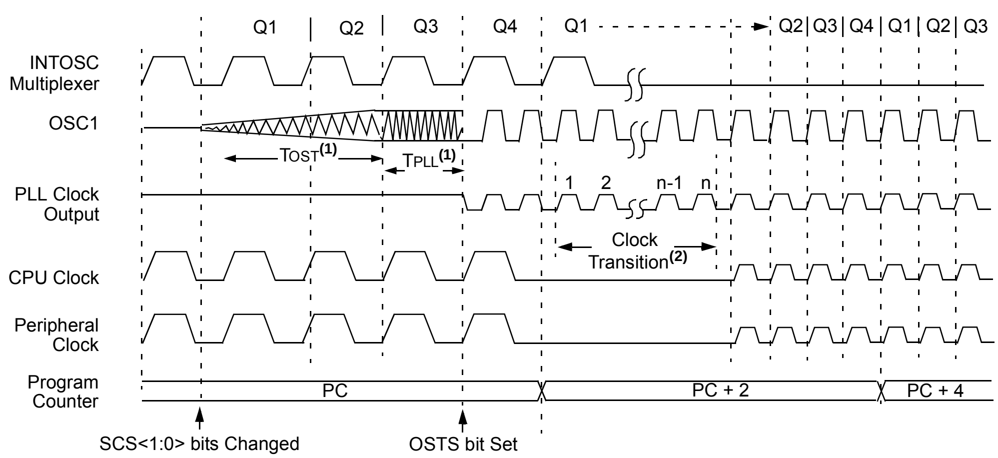
On transitions from RC_RUN mode to PRI_RUN mode, the device continues to be clocked from the INTOSC multiplexer while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 4-4). When the clock switch is complete, the HFIOFS or MFIOFS bit is cleared, the OSTS bit is set and the primary clock is providing the device clock. The IDLEN and SCS bits are not affected by the switch. The LF-INTOSC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

FIGURE 4-3: TRANSITION TIMING TO RC_RUN MODE



Note 1: Clock transition typically occurs within 2-4 TOSC.

FIGURE 4-4: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

2: Clock transition typically occurs within 2-4 TOSC.

REGISTER 4-4: PMD0: PERIPHERAL MODULE DISABLE REGISTER 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSP2MD	SSP1MD	ADCMD
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 **CCP3MD:** PMD ECCP3 Enable/Disable bit
1 = Peripheral Module Disable (PMD) is enabled for ECCP3, disabling all of its clock sources
0 = PMD is disabled for ECCP3

bit 6 **CCP2MD:** PMD ECCP2 Enable/Disable bit
1 = PMD is enabled for ECCP2, disabling all of its clock sources
0 = PMD is disabled for ECCP2

bit 5 **CCP1MD:** PMD ECCP1 Enable/Disable bit
1 = PMD is enabled for ECCP1, disabling all of its clock sources
0 = PMD is disabled for ECCP1

bit 4 **UART2MD:** PMD UART2 Enable/Disable bit
1 = PMD is enabled for UART2, disabling all of its clock sources
0 = PMD is disabled for UART2

bit 3 **UART1MD:** PMD UART1 Enable/Disable bit
1 = PMD is enabled for UART1, disabling all of its clock sources
0 = PMD is disabled for UART1

bit 2 **SSP2MD:** PMD MSSP2 Enable/Disable bit
1 = PMD is enabled for MSSP2, disabling all of its clock sources
0 = PMD is disabled for MSSP2

bit 1 **SSP1MD:** PMD MSSP1 Enable/Disable bit
1 = PMD is enabled for MSSP1, disabling all of its clock sources
0 = PMD is disabled for MSSP1

bit 0 **ADCMD:** PMD Analog/Digital Converter PMD Enable/Disable bit
1 = PMD is enabled for the Analog/Digital Converter, disabling all of its clock sources
0 = PMD is disabled for the Analog/Digital Converter

4.7 Ultra Low-Power Wake-up

The Ultra Low-Power Wake-up (ULPWU) on pin, RA0, allows a slow falling voltage to generate an interrupt without excess current consumption.

To use this feature:

1. Charge the capacitor on RA0 by configuring the RA0 pin to an output and setting it to ‘1’.
2. Stop charging the capacitor by configuring RA0 as an input.
3. Discharge the capacitor by setting the ULPEN and ULPSINK bits in the WDTCON register.
4. Configure Sleep mode.
5. Enter Sleep mode.

When the voltage on RA0 drops below V_{IL}, the device wakes up and executes the next instruction.

This feature provides a low-power technique for periodically waking up the device from Sleep mode.

The time-out is dependent on the discharge time of the RC circuit on RA0.

When the ULPWU module wakes the device from Sleep mode, the ULPLVL bit (WDTCON<5>) is set. Software can check this bit upon wake-up to determine the wake-up source.

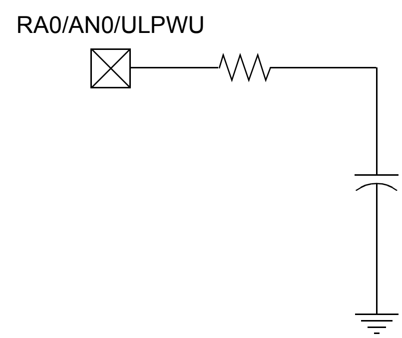
See Example 4-1 for initializing the ULPWU module.

EXAMPLE 4-1: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
           //***************************
           //Charge the capacitor on RA0
           //***************************
TRISAbits.TRISA0 = 0;
PORTAbits.RA0 = 1;
for(i = 0; i < 10000; i++) Nop();
           //*****************************
           //Stop Charging the capacitor
           //on RA0
           //*****************************
TRISAbits.TRISA0 = 1;
           //*****************************
           //Enable the Ultra Low Power
           //Wakeup module and allow
           //capacitor discharge
           //*****************************
WDTCONbits.ULPEN = 1;
WDTCONbits.ULPSINK = 1;
           //For Sleep
OSCCONbits.IDLEN = 0;
           //Enter Sleep Mode
           //
Sleep();
           //for sleep, execution will
           //resume here
```

A series resistor, between RA0 and the external capacitor, provides overcurrent protection for the RA0/AN0/ULPWU pin and enables software calibration of the time-out (see Figure 4-9).

FIGURE 4-9: ULTRA LOW-POWER WAKE-UP INITIALIZATION



A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired delay in Sleep. This technique compensates for the affects of temperature, voltage and component accuracy. The peripheral can also be configured as a simple Programmable Low-Voltage Detect (LVD) or temperature sensor.

> **Note:** For more information, see *AN879, “Using the Microchip Ultra Low-Power Wake-up Module”* (DS00879).

5.7 Reset State of Registers

Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a "Reset state" depending on the type of Reset that occurred.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register ($\overline{\text{CM}}$, $\overline{\text{RI}}$, $\overline{\text{TO}}$, $\overline{\text{PD}}$, $\overline{\text{POR}}$ and $\overline{\text{BOR}}$) are set or cleared differently in different Reset situations, as indicated in Table 5-1. These bits are used in software to determine the nature of the Reset.

Table 5-2 describes the Reset states for all of the Special Function Registers. These are categorized by Power-on and Brown-out Resets, Master Clear and WDT Resets, and WDT wake-ups.

TABLE 5-1: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter[1]	RCON Register						STKPTR Register	
		$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	STKFUL	STKUNF
Power-on Reset	0000h	1	1	1	1	0	0	0	0
RESET instruction	0000h	u	0	u	u	u	u	u	u
Brown-out Reset	0000h	1	1	1	1	u	0	u	u
Configuration Mismatch Reset	0000h	0	u	u	u	u	u	u	u
$\overline{\text{MCLR}}$ Reset during power-managed Run modes	0000h	u	u	1	u	u	u	u	u
$\overline{\text{MCLR}}$ Reset during power-managed Idle modes and Sleep mode	0000h	u	u	1	0	u	u	u	u
$\overline{\text{MCLR}}$ Reset during full-power execution	0000h	u	u	u	u	u	u	u	u
Stack Full Reset (STVREN = 1)	0000h	u	u	u	u	u	u	1	u
Stack Underflow Reset (STVREN = 1)	0000h	u	u	u	u	u	u	u	1
Stack Underflow Error (not an actual Reset, STVREN = 0)	0000h	u	u	u	u	u	u	u	1
WDT time-out during full-power or power-managed Run modes	0000h	u	u	0	u	u	u	u	u
WDT time-out during power-managed Idle or Sleep modes	PC + 2	u	u	0	0	u	u	u	u
Interrupt exit from power-managed modes	PC + 2	u	u	u	0	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

7.2.2 TABLAT – TABLE LATCH REGISTER

The Table Latch (TABLAT) is an eight-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

7.2.3 TBLPTR – TABLE POINTER REGISTER

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the Device ID, the User ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the `TBLRD` and `TBLWT` instructions. These instructions can update the TBLPTR in one of four ways, based on the table operation. These operations are shown in Table 7-1 and only affect the low-order 21 bits.

7.2.4 TABLE POINTER BOUNDARIES

The TBLPTR is used in reads, writes and erases of the Flash program memory.

When a `TBLRD` is executed, all 22 bits of the TBLPTR determine which byte is read from program memory into the TABLAT.

When a `TBLWT` is executed, the six LSbs of the Table Pointer register (TBLPTR<5:0>) determine which of the 64 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 16 MSbs of the TBLPTR (TBLPTR<21:6>) determine which program memory block of 64 bytes is written to. For more detail, see **Section 7.5 “Writing to Flash Program Memory”**.

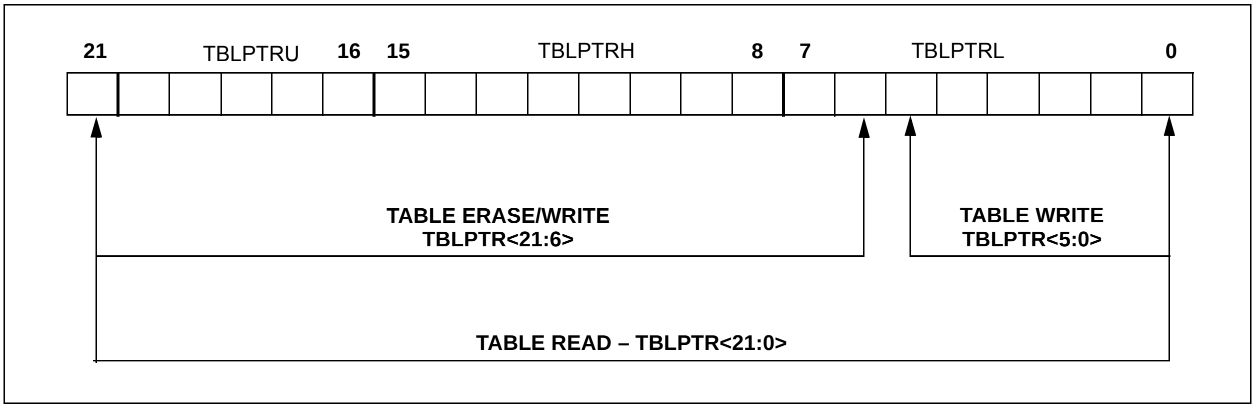
When an erase of program memory is executed, the 16 MSbs of the Table Pointer register (TBLPTR<21:6>) point to the 64-byte block that will be erased. The Least Significant bits (TBLPTR<5:0>) are ignored.

Figure 7-3 describes the relevant boundaries of the TBLPTR based on Flash program memory operations.

TABLE 7-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
`TBLRD*` `TBLWT*`	TBLPTR is not modified
`TBLRD*+` `TBLWT*+`	TBLPTR is incremented after the read/write
`TBLRD*-` `TBLWT*-`	TBLPTR is decremented after the read/write
`TBLRD+*` `TBLWT+*`	TBLPTR is incremented before the read/write

FIGURE 7-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



7.5 Writing to Flash Program Memory

The programming blocks are:

- PIC18FX5K22 and PIC18FX6K22 – 32 words or 64 bytes
- PIC18FX7K22 – 64 words or 128 bytes

Word or byte programming is not supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. The number of holding registers used for programming by the table writes are:

- PIC18FX5K22 and PIC18FX6K22 – 64
- PIC18FX7K22 – 128

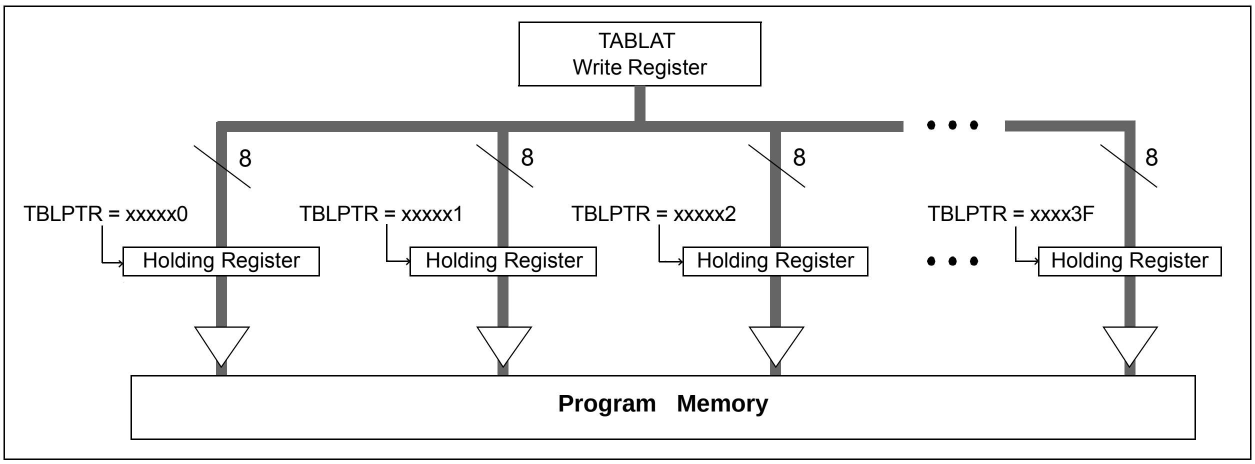
Since the Table Latch (TABLAT) is only a single byte, the `TBLWT` instruction may need to be executed 64 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 or 128 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

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

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

Note:	The default value of the holding registers on device Resets, and after write operations, is FFh. A write of FFh to a holding register does not modify that byte. This means that individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '`0`' to a '`1`'. When modifying individual bytes, it is not necessary to load all 64 or 128 holding registers before executing a write operation.

FIGURE 7-5: TABLE WRITES TO FLASH PROGRAM MEMORY



Example 10-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 10-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES3:RES0).

EQUATION 10-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
\text{RES3:RES0} &= \text{ARG1H:ARG1L} \bullet \text{ARG2H:ARG2L} \\
&= (\text{ARG1H} \bullet \text{ARG2H} \bullet 2^{16}) + \\
&\quad (\text{ARG1H} \bullet \text{ARG2L} \bullet 2^{8}) + \\
&\quad (\text{ARG1L} \bullet \text{ARG2H} \bullet 2^{8}) + \\
&\quad (\text{ARG1L} \bullet \text{ARG2L})
\end{aligned}
$$

EXAMPLE 10-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 10-4 shows the sequence to do a 16 x 16 signed multiply. Equation 10-2 shows the algorithm used. The 32-bit result is stored in four registers (RES3:RES0). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

EQUATION 10-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
\text{RES3:RES0} &= \text{ARG1H:ARG1L} \bullet \text{ARG2H:ARG2L} \\
&= (\text{ARG1H} \bullet \text{ARG2H} \bullet 2^{16}) + \\
&\quad (\text{ARG1H} \bullet \text{ARG2L} \bullet 2^{8}) + \\
&\quad (\text{ARG1L} \bullet \text{ARG2H} \bullet 2^{8}) + \\
&\quad (\text{ARG1L} \bullet \text{ARG2L}) + \\
&\quad (-1 \bullet \text{ARG2H<7>} \bullet \text{ARG1H:ARG1L} \bullet 2^{16}) + \\
&\quad (-1 \bullet \text{ARG1H<7>} \bullet \text{ARG2H:ARG2L} \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 10-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
   MOVF    ARG1L, W
   MULWF   ARG2L        ; ARG1L * ARG2L ->
                        ; PRODH:PRODL
   MOVFF   PRODH, RES1 ;
   MOVFF   PRODL, RES0 ;
;
   MOVF    ARG1H, W
   MULWF   ARG2H        ; ARG1H * ARG2H ->
                        ; PRODH:PRODL
   MOVFF   PRODH, RES3 ;
   MOVFF   PRODL, RES2 ;
;
   MOVF    ARG1L, W
   MULWF   ARG2H        ; ARG1L * ARG2H ->
                        ; PRODH:PRODL
   MOVF    PRODL, W     ;
   ADDWF   RES1, F      ; Add cross
   MOVF    PRODH, W     ; products
   ADDWFC  RES2, F      ;
   CLRF    WREG         ;
   ADDWFC  RES3, F      ;
;
   MOVF    ARG1H, W     ;
   MULWF   ARG2L        ; ARG1H * ARG2L ->
                        ; PRODH:PRODL
   MOVF    PRODL, W     ;
   ADDWF   RES1, F      ; Add cross
   MOVF    PRODH, W     ; products
   ADDWFC  RES2, F      ;
   CLRF    WREG         ;
   ADDWFC  RES3, F      ;
;
   BTFSS   ARG2H, 7     ; ARG2H:ARG2L neg?
   BRA     SIGN_ARG1    ; no, check ARG1
   MOVF    ARG1L, W     ;
   SUBWF   RES2         ;
   MOVF    ARG1H, W     ;
   SUBWFB  RES3         ; SIGN_ARG1
   BTFSS   ARG1H, 7     ; ARG1H:ARG1L neg?
   BRA     CONT_CODE    ; no, done
   MOVF    ARG2L, W     ;
   SUBWF   RES2         ;
   MOVF    ARG2H, W     ;
   SUBWFB  RES3
;
CONT_CODE
   :
```

12.4 PORTC, TRISC and LATC Registers

PORTC is an eight-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISC and LATC. Only PORTC pins, RC2 through RC7, are digital only pins.

PORTC is multiplexed with ECCP, MSSP and EUSART peripheral functions (Table 12-5). The pins have Schmitt Trigger input buffers. The pins for ECCP, SPI and EUSART are also configurable for open-drain output whenever these functions are active. Open-drain configuration is selected by setting the SPIOD, CCPxOD and U1OD control bits in the registers, ODCON1 and ODCON3.

RC1 is normally configured as the default peripheral pin for the ECCP2 module. The assignment of ECCP2 is controlled by Configuration bit, CCP2MX (default state, CCP2MX = 1).

When enabling peripheral functions, use care in defining TRIS bits for each PORTC pin. Some peripherals can override the TRIS bit to make a pin an output or input. Consult the corresponding peripheral section for the correct TRIS bit settings.

> **Note:** These pins are configured as digital inputs on any device Reset.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

EXAMPLE 12-3: INITIALIZING PORTC

```
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

TABLE 12-5: PORTC FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RC0/SOSCO/SCLKI/	RC0	0	O	DIG	LATC<0> data output.
		1	I	ST	PORTC<0> data input.
	SOSCO	1	I	ST	SOSC oscillator output.
	SCLKI	1	I	ST	Digital clock input; enabled when SOSC oscillator is disabled.
RC1/SOSCI/ECCP2/P2A	RC1	0	O	DIG	LATC<1> data output.
		1	I	ST	PORTC<1> data input.
	SOSCI	x	I	ANA	SOSC oscillator input.
	ECCP2(1)	0	O	DIG	ECCP2 compare output and ECCP2 PWM output; takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A	0	O	DIG	ECCP2 Enhanced PWM output, Channel A. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.
RC2/ECCP1/P1A	RC2	0	O	DIG	LATC<2> data output.
		1	I	ST	PORTC<2> data input.
	ECCP1	0	O	DIG	ECCP1 compare output and ECCP1 PWM output; takes priority over port data.
		1	I	ST	ECCP1 capture input.
	P1A	0	O	DIG	ECCP1 Enhanced PWM output, Channel A. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, TTL = TTL Buffer Input, I^{2}C = I^{2}C™/SMBus Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Default assignment for ECCP2 when the CCP2MX Configuration bit is set.

16.3 Timer3/5/7 16-Bit Read/Write Mode

Timer3/5/7 can be configured for 16-bit reads and writes (see Figure 16.3). When the RD16 control bit (TxCON<1>) is set, the address for TMRxH is mapped to a buffer register for the high byte of Timer3/5/7. A read from TMRxL will load the contents of the high byte of Timer3/5/7 into the Timerx High Byte Buffer register. This provides users with the ability to accurately read all 16 bits of Timer3/5/7 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer3/5/7 must also take place through the TMRxH Buffer register. The Timer3/5/7 high byte is updated with the contents of TMRxH when a write occurs to TMRxL. This allows users to write all 16 bits to both the high and low bytes of Timer3/5/7 at once.

The high byte of Timer3/5/7 is not directly readable or writable in this mode. All reads and writes must take place through the Timerx High Byte Buffer register.

Writes to TMRxH do not clear the Timer3/5/7 prescaler. The prescaler is only cleared on writes to TMRxL.

16.4 Using the SOSC Oscillator as the Timer3/5/7 Clock Source

The SOSC internal oscillator may be used as the clock source for Timer3/5/7. The SOSC oscillator is enabled by any peripheral that requests it. There are eight ways the SOSC can be enabled: if the SOSC is selected as the source by any of the odd timers, which is done by each respective SOSCEN bit (TxCON<3>), if the SOSC is selected as the RTCC source by the RTCOSC Configuration bit (CONFIG3L<1>), if the SOSC is selected as the CPU clock source by the SCS bits (OSCCON<1:0>) or if the SOSCGO bit is set (OSCCON2<3>). The SOSCGO bit is used to warm up the SOSC so that it is ready before any peripheral requests it. To use it as the Timer3/5/7 clock source, the TMRxCS bit must also be set. As previously noted, this also configures Timer3/5/7 to increment on every rising edge of the oscillator source.

The SOSC oscillator is described in **Section 14.5 "SOSC Oscillator"**.

REGISTER 18-8: MONTH: MONTH VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
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 7-5 **Unimplemented:** Read as ‘0’

bit 4 **MTHTEN0:** Binary Coded Decimal Value of Month’s Tens Digit bit
Contains a value of 0 or 1.

bit 3-0 **MTHONE<3:0>:** Binary Coded Decimal Value of Month’s Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 18-9: DAY: DAY VALUE REGISTER(1)

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal value of Day’s Tens Digit bits
Contains a value from 0 to 3.

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day’s Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 18-10: WEEKDAY: WEEKDAY VALUE REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
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 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

Note 1: A write to this register is only allowed when RTCWREN = 1.

19.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (generically, CCPxCON) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte). All registers are both readable and writable.

19.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers, 1 through 8, which vary with the selected mode. Various timers are available to the CCP modules in Capture, Compare or PWM modes, as shown in Table 19-1.

TABLE 19-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1, Timer3, Timer 5 or Timer7
Compare	
PWM	Timer2, Timer4, Timer 6 or Timer8

The assignment of a particular timer to a module is determined by the timer to CCP enable bits in the CCPTMRSx registers (see Register 19-2 and Register 19-3). All of the modules may be active at once and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time.

The CCPTMRS1 register selects the timers for CCP modules, 7, 6, 5 and 4, and the CCPTMRS2 register selects the timers for CCP modules, 10, 9 and 8. The possible configurations are shown in Table 19-2 and Table 19-3.

TABLE 19-2: TIMER ASSIGNMENTS FOR CCP MODULES 4, 5, 6 AND 7

CCPTMRS1 Register											
CCP4			CCP5			CCP6			CCP7		
C4TSEL<1:0>	Capture/ Compare Mode	PWM Mode	C5TSEL0	Capture/ Compare Mode	PWM Mode	C6TSEL0	Capture/ Compare Mode	PWM Mode	C7TSEL<1:0>	Capture/ Compare Mode	PWM Mode
0 0	TMR1	TMR2	0	TMR1	TMR2	0	TMR1	TMR2	0 0	TMR1	TMR2
0 1	TMR3	TMR4	1	TMR5	TMR4	1	TMR5	TMR2	0 1	TMR5	TMR4
1 0	TMR3	TMR6							1 0	TMR5	TMR6
1 1	Reserved[(1)]								1 1	TMR5	TMR8

Note 1: Do not use the reserved bits.

TABLE 19-3: TIMER ASSIGNMENTS FOR CCP MODULES 8, 9 AND 10

CCPTMRS2 Register											
CCP8			CCP8 Devices with 32 Kbytes			CCP9[(1)]			CCP10[(1)]		
C8TSEL<1:0>	Capture/ Compare Mode	PWM Mode	C8TSEL<1:0>	Capture/ Compare Mode	PWM Mode	C9TSEL0	Capture/ Compare Mode	PWM Mode	C10TSEL0	Capture/ Compare Mode	PWM Mode
0 0	TMR1	TMR2	0 0	TMR1	TMR2	0	TMR1	TMR2	0	TMR1	TMR2
0 1	TMR7	TMR4	0 1	TMR1	TMR4	1	TMR7	TMR4	1	TMR7	TMR2
1 0	TMR7	TMR6	1 0	TMR1	TMR6						
1 1	Reserved[(2)]		1 1	Reserved[(2)]							

Note 1: The module is not available for devices with 32 Kbytes of program memory (PIC18F65K22 and PIC18F85K22).

2: Do not use the reserved setting.

TABLE 28-4: SUMMARY OF CODE PROTECTION REGISTERS

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	CP7(1)	CP6(1)	CP5(1)	CP4(1)	CP3	CP2	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	WRT7(1)	WRT6(1)	WRT5(1)	WRT4(1)	WRT3	WRT2	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	EBTR7(1)	EBTR6(1)	EBTR5(1)	EBTR4(1)	EBTR3	EBTR2	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

Legend: Shaded cells are unimplemented.

Note 1: This bit is available only on the PIC18F67K22 and PIC18F87K22 devices.

28.6.1 PROGRAM MEMORY CODE PROTECTION

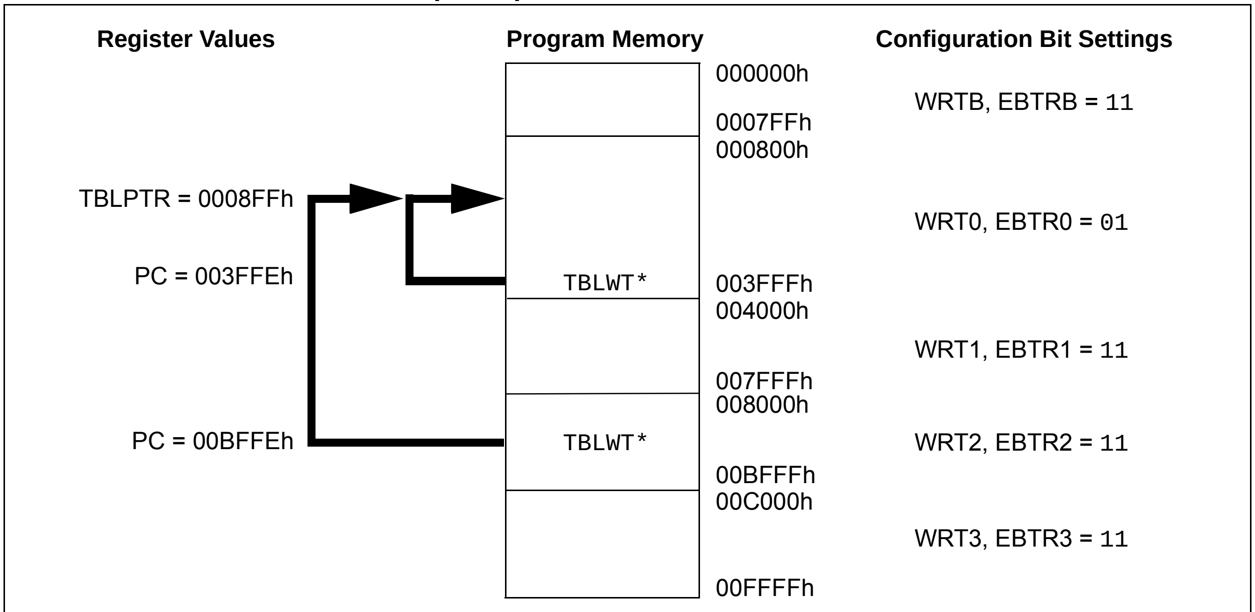
The program memory may be read to, or written from, any location using the table read and table write instructions. The Device ID may be read with table reads. The Configuration registers may be read and written with the table read and table write instructions.

In normal Execution mode, the CPx bits have no direct effect. CPx bits inhibit external reads and writes. A block of user memory may be protected from table writes if the WRTx Configuration bit is '0'.

The EBTRx bits control table reads. For a block of user memory, with the EBTRx bit set to '0', a table read instruction that executes from within that block is allowed to read. A table read instruction that executes from a location outside of that block is not allowed to read and will result in reading '0's. Figures 28-7 through 28-9 illustrate table write and table read protection.

> **Note:** Code protection bits may only be written to a '0' from a '1' state. It is not possible to write a '1' to a bit in the '0' state. Code protection bits are only set to '1' by a full chip erase or block erase function. The full chip erase and block erase functions can only be initiated via ICSP or an external programmer. Refer to the device programming specification for more information.

FIGURE 28-7: TABLE WRITE (WRTx) DISALLOWED



Results: All table writes are disabled to Blockn whenever WRTx = 0.

CLRF **Clear f**

Syntax:	CLRF f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	000h → f, 1 → Z
Status Affected:	Z
Encoding:	`0110` `101a` `ffff` `ffff`
Description:	Clears the contents of the specified register. If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank. If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 29.2.3 “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: `CLRF FLAG_REG,1`

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWDT **Clear Watchdog Timer**

Syntax:	CLRWDT
Operands:	None
Operation:	000h → WDT, 000h → WDT postscaler, 1 → $\overline{\text{TO}}$, 1 → $\overline{\text{PD}}$
Status Affected:	$\overline{\text{TO}}$, $\overline{\text{PD}}$
Encoding:	`0000` `0000` `0000` `0100`
Description:	`CLRWDT` instruction resets the Watchdog Timer. It also resets the post-scaler of the WDT. Status bits, $\overline{\text{TO}}$ and $\overline{\text{PD}}$, are set.
Words:	1
Cycles:	1

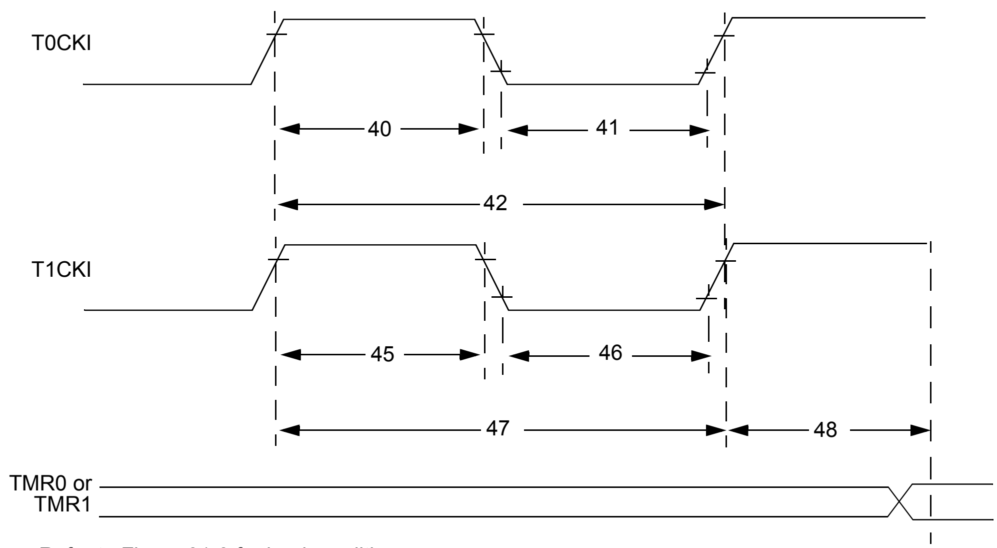
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: `CLRWDT`

Before Instruction
WDT Counter = ?
After Instruction
WDT Counter = 00h
WDT Postscaler = 0
$\overline{\text{TO}}$ = 1
$\overline{\text{PD}}$ = 1

FIGURE 31-12: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



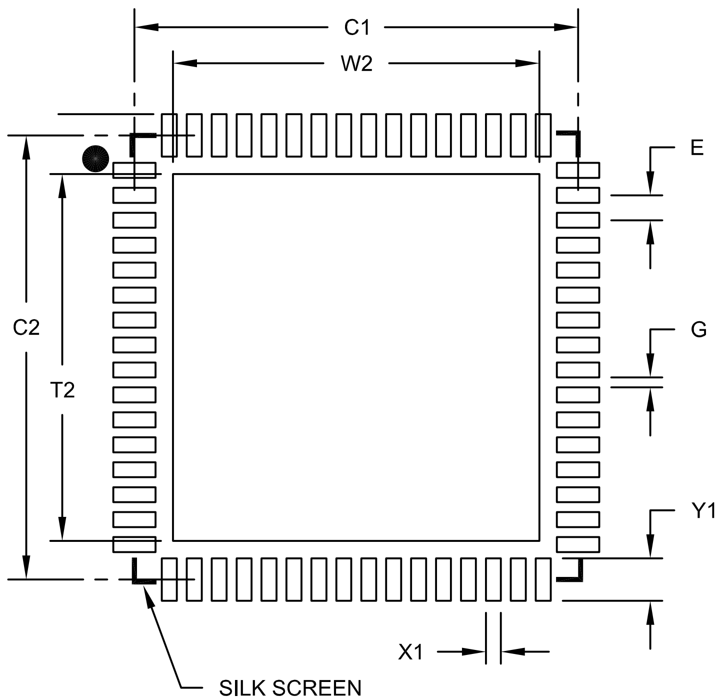
Note: Refer to Figure 31-3 for load conditions.

TABLE 31-15: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	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	T1CKI High Time	Synchronous, no prescaler	0.5 TCY + 20	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
46	TT1L	T1CKI Low Time	Synchronous, no prescaler	0.5 TCY + 5	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
47	TT1P	T1CKI Input Period	Synchronous	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	ns	
	FT1	T1CKI Oscillator Input Frequency Range		DC	50	kHz	
48	TCKE2TMRI	Delay from External T1CKI Clock Edge to Timer Increment		2 TOSC	7 TOSC	—	

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]
With 0.40 mm Contact Length

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



RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	W2			7.35
Optional Center Pad Length	T2			7.35
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2149A