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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f23k22-i-ss

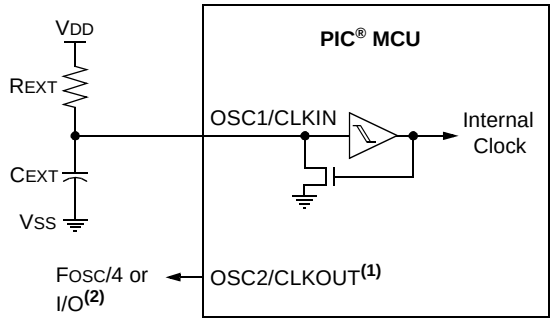
2.5.4 EXTERNAL RC MODES

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes: RC and RCIO.

2.5.4.1 RC Mode

In RC mode, the RC circuit connects to OSC1. OSC2/CLKOUT outputs the RC oscillator frequency divided by four. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 2-8 shows the external RC mode connections.

FIGURE 2-8: EXTERNAL RC MODES



Recommended values: 10 kΩ ≤ REXT ≤ 100 kΩ
CEXT > 20 pF

Note 1: Alternate pin functions are listed in **Section 1.0 "Device Overview"**.
2: Output depends upon RC or RCIO clock mode.

2.5.4.2 RCIO Mode

In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes a general purpose I/O pin.

The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- input threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

2.6 Internal Clock Modes

The oscillator module has three independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 2-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 2-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory calibrated and operates at 31.25 kHz. The LFINTOSC cannot be user-adjusted, but is designed to be stable over temperature and voltage.

The system clock speed can be selected via software using the Internal Oscillator Frequency select bits IRCF<2:0> of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS<1:0>) bits of the OSCCON register. See **Section 2.11 "Clock Switching"** for more information.

2.6.1 INTOSC WITH I/O OR CLOCKOUT

Two of the clock modes selectable with the FOSC<3:0> bits of the CONFIG1H Configuration register configure the internal oscillator block as the primary oscillator. Mode selection determines whether the OSC2/CLKOUT pin will be configured as general purpose I/O or FOSC/4 (CLKOUT). In both modes, the OSC1/CLKIN pin is configured as general purpose I/O. See **Section 24.0 "Special Features of the CPU"** for more information.

The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

2.9 Effects of Power-Managed Modes on the Various Clock Sources

For more information about the modes discussed in this section see **Section 3.0 "Power-Managed Modes"**. A quick reference list is also available in Table 3-1.

When PRI_IDLE mode is selected, the designated primary oscillator continues to run without interruption. For all other power-managed modes, the oscillator using the OSC1 pin is disabled. The OSC1 pin (and OSC2 pin, if used by the oscillator) will stop oscillating.

In secondary clock modes (SEC_RUN and SEC_IDLE), the secondary oscillator (SOSC) is operating and providing the device clock. The secondary oscillator may also run in all power-managed modes if required to clock Timer1, Timer3 or Timer5.

In internal oscillator modes (INTOSC_RUN and INTOSC_IDLE), the internal oscillator block provides the device clock source. The 31.25 kHz LFINTOSC output can be used directly to provide the clock and may be enabled to support various special features, regardless of the power-managed mode (see **Section 24.3 "Watchdog Timer (WDT)"**, **Section 2.12 "Two-Speed Clock Start-up Mode"** and **Section 2.13 "Fail-Safe Clock Monitor"** for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up). The HFINTOSC and MFINTOSC outputs may be used directly to clock the device or may be divided down by the postscaler. The HFINTOSC and MFINTOSC outputs are disabled when the clock is provided directly from the LFINTOSC output.

When the Sleep mode is selected, all clock sources are stopped. Since all the transistor switching currents have been stopped, Sleep mode achieves the lowest current consumption of the device (only leakage currents).

Enabling any on-chip feature that will operate during Sleep will increase the current consumed during Sleep. The LFINTOSC is required to support WDT operation. Other features may be operating that do not require a device clock source (i.e., SSP slave, PSP, INTn pins and others). Peripherals that may add significant current consumption are listed in **Section 27.8 "DC Characteristics: Input/Output Characteristics, PIC18(L)F2X/4XK22".**

2.10 Power-up Delays

Power-up delays are controlled by two timers, so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 4.6 "Device Reset Timers"**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up. It is enabled by clearing (= 0) the PWRTEN Configuration bit.

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (LP, XT and HS modes). The OST does this by counting 1024 oscillator cycles before allowing the oscillator to clock the device.

When the PLL is enabled with external oscillator modes, the device is kept in Reset for an additional 2 ms, following the OST delay, so the PLL can lock to the incoming clock frequency.

There is a delay of interval T_{CSD}, following POR, while the controller becomes ready to execute instructions. This delay runs concurrently with any other delays. This may be the only delay that occurs when any of the EC, RC or INTIOSC modes are used as the primary clock source.

When the HFINTOSC is selected as the primary clock, the main system clock can be delayed until the HFINTOSC is stable. This is user selectable by the HFOFST bit of the CONFIG3H Configuration register. When the HFOFST bit is cleared, the main system clock is delayed until the HFINTOSC is stable. When the HFOFST bit is set, the main system clock starts immediately.

In either case, the HFIOFS bit of the OSCCON register can be read to determine whether the HFINTOSC is operating and stable.

REGISTER 3-2: PMD1: PERIPHERAL MODULE DISABLE REGISTER 1

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
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 **MSSP2MD:** MSSP2 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 6 **MSSP1MD:** MSSP1 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 5 **Unimplemented:** Read as '0'

bit 4 **CCP5MD:** CCP5 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 3 **CCP4MD:** CCP4 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 2 **CCP3MD:** CCP3 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 1 **CCP2MD:** CCP2 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 0 **CCP1MD:** CCP1 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

5.4.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Block 15. The lower half is known as the “Access RAM” and is composed of GPRs. This upper half is also where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figures 5-5 through 5-7).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘1’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘0’, however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle, without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 5.7.3 “Mapping the Access Bank in Indexed Literal Offset Mode”**.

5.4.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM, which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

5.4.4 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. SFRs start at the top of data memory (FFFh) and extend downward to occupy the top portion of Bank 15 (F38h to FFFh). A list of these registers is given in Table 5-1 and Table 5-2.

The SFRs can be classified into two sets: those associated with the “core” device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and interrupt registers are described in their respective chapters, while the ALU’s STATUS register is described later in this section. Registers related to the operation of a peripheral feature are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as ‘0’s.

5.6.3.1 FSR Registers and the INDF Operand

At the core of indirect addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. Each FSR pair holds a 12-bit value, therefore, the four upper bits of the FSRnH register are not used. The 12-bit FSR value can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as “virtual” registers: they are mapped in the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction’s target. The INDF operand is just a convenient way of using the pointer.

Because indirect addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

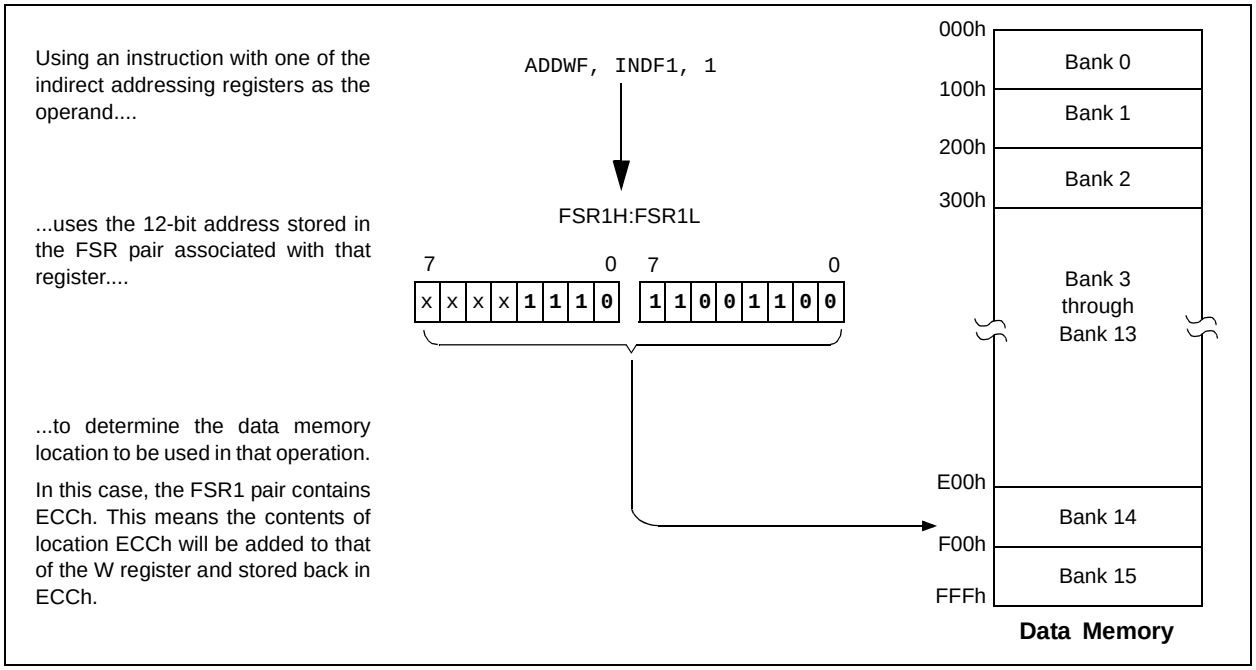
5.6.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers which cannot be directly read or written. Accessing these registers actually accesses the location to which the associated FSR register pair points, and also performs a specific action on the FSR value. They are:

- POSTDEC: accesses the location to which the FSR points, then automatically decrements the FSR by 1 afterwards
- POSTINC: accesses the location to which the FSR points, then automatically increments the FSR by 1 afterwards
- PREINC: automatically increments the FSR by one, then uses the location to which the FSR points 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 location to which the result points in the operation.

In this context, accessing an INDF register uses the value in the associated FSR register without changing it. Similarly, accessing a PLUSW register gives the FSR value an offset by that in the W register; however, neither W nor the FSR is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR register.

FIGURE 5-10: INDIRECT ADDRESSING



7.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit of the EECON1 register and then set control bit, RD. The data is available on the very next instruction cycle; therefore, the EEDATA register can be read by the next instruction. EEDATA will hold this value until another read operation, or until it is written to by the user (during a write operation).

The basic process is shown in Example 7-1.

7.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. The sequence in Example 7-2 must be followed to initiate the write cycle.

The write will not begin if this sequence is not exactly followed (write 55h to EECON2, write 0AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared by hardware and the EEPROM Interrupt Flag bit, EEIF, is set. The user may either enable this interrupt or poll this bit. EEIF must be cleared by software.

7.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

EXAMPLE 7-1: DATA EEPROM READ

```
        MOVLW   DATA_EE_ADDR   ;
        MOVWF   EEADR          ; Data Memory Address to read
        BCF     EECON1, EEPGD  ; Point to DATA memory
        BCF     EECON1, CFGS   ; Access EEPROM
        BSF     EECON1, RD     ; EEPROM Read
        MOVF    EEDATA, W      ; W = EEDATA
```

EXAMPLE 7-2: DATA EEPROM WRITE

```
                MOVLW   DATA_EE_ADDR_LOW   ;
                MOVWF   EEADR              ; Data Memory Address to write
                MOVLW   DATA_EE_ADDR_HI    ;
                MOVWF   EEADRH             ;
                MOVLW   DATA_EE_DATA       ;
                MOVWF   EEDATA             ; Data Memory Value to write
                BCF     EECON1, EEPGD      ; Point to DATA memory
                BCF     EECON1, CFGS       ; Access EEPROM
                BSF     EECON1, WREN       ; Enable writes
                BCF     INTCON, GIE        ; Disable Interrupts
                MOVLW   55h                ;
Required        MOVWF   EECON2             ; Write 55h
Sequence        MOVLW   0AAh               ;
                MOVWF   EECON2             ; Write 0AAh
                BSF     EECON1, WR         ; Set WR bit to begin write
                BSF     INTCON, GIE        ; Enable Interrupts

                                           ; User code execution
                BCF     EECON1, WREN       ; Disable writes on write complete (EEIF set)
```

10.4 PORTC Registers

PORTC is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC.

PORTC is multiplexed with several peripheral functions (Table 10-8). The pins have Schmitt Trigger input buffers.

Some of these pin functions can be relocated to alternate pins using the Control fuse bits in CONFIG3H. RC0 is the default pin for T3CKI. Clearing the T3CMX bit moves the pin function to RB5. RC1 is the default pin for the CCP2 peripheral pin. Clearing the CCP2MX bit moves the pin function to the RB3 pin.

Two other pin functions, P2B and CCP3, can be relocated from their default pins to PORTC pins by clearing the control fuses in CONFIG3H. Clearing P2BMX and CCP3MX moves the pin functions to RC0 and RC6**(1)**/RE0**(2)**, respectively.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. The EUSART and MSSP peripherals override the TRIS bit to make a pin an output or an input, depending on the peripheral configuration. Refer to the corresponding peripheral section for additional information.

Note:	On a Power-on Reset, these pins are configured as analog inputs.

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 10-3: INITIALIZING PORTC

```
MOVLB   0xF     ; Set BSR for banked SFRs
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
MOVLW   30h     ; Value used to
                ; enable digital inputs
MOVWF   ANSELC  ; RC<3:2> dig input enable
                ; No ANSEL bits for RC<1:0>
                ; RC<7:6> dig input enable
```

10.4.1 PORTC OUTPUT PRIORITY

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTC pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

10.7 Port Analog Control

Most port pins are multiplexed with analog functions such as the Analog-to-Digital Converter and comparators. When these I/O pins are to be used as analog inputs it is necessary to disable the digital input buffer to avoid excessive current caused by improper biasing of the digital input. Individual control of the digital input buffers on pins which share analog functions is provided by the ANSELA, ANSELB, ANSELC, ANSELD and ANSELE registers. Setting an ANSx bit high will disable the associated digital input buffer and cause all reads of that pin to return '0' while allowing analog functions of that pin to operate correctly.

The state of the ANSx bits has no affect on digital output functions. A pin with the associated TRISx bit clear and ANSx bit set will still operate as a digital output but the input mode will be analog. This can cause unexpected behavior when performing read-modify-write operations on the affected port.

All ANSEL register bits default to '1' upon POR and BOR, disabling digital inputs for their associated port pins. All TRIS register bits default to '1' upon POR or BOR, disabling digital outputs for their associated port pins. As a result, all port pins that have an ANSEL register will default to analog inputs upon POR or BOR.

10.8 Port Slew Rate Control

The output slew rate of each port is programmable to select either the standard transition rate or a reduced transition rate of approximately 0.1 times the standard to minimize EMI. The reduced transition time is the default slew rate for all ports.

10.9 Register Definitions – Port Control

REGISTER 10-1: PORTX(1): PORTx REGISTER

R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x
Rx7	Rx6	Rx5	Rx4	Rx3	Rx2	Rx1	Rx0
bit 7							bit 0

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

bit 7-0 **Rx<7:0>:** PORTx I/O bit values(2)

Note 1: Register Description for PORTA, PORTB, PORTC and PORTD.

2: Writes to PORTx are written to corresponding LATx register. Reads from PORTx register is return of I/O pin values.

REGISTER 10-10: LATx: PORTx OUTPUT LATCH REGISTER(1)

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATx7	LATx6	LATx5	LATx4	LATx3	LATx2	LATx1	LATx0
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 7-0 **LATx<7:0>:** PORTx Output Latch bit value(2)

Note 1: Register Description for LATA, LATB, LATC and LATD.

2: Writes to PORTx are written to corresponding LATx register. Reads from PORTx register is return of I/O pin values.

REGISTER 10-11: LATE: PORTE OUTPUT LATCH REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	LATE2	LATE1	LATE0
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 **LATE<2:0>:** PORTE Output Latch bit value(2)

Note 1: Available on PIC18(L)F4XK22 devices only.

2: Writes to PORTE are written to corresponding LATE register. Reads from PORTE register is return of I/O pin values.

REGISTER 10-12: WPUB: WEAK PULL-UP PORTB REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
bit 7							bit 0

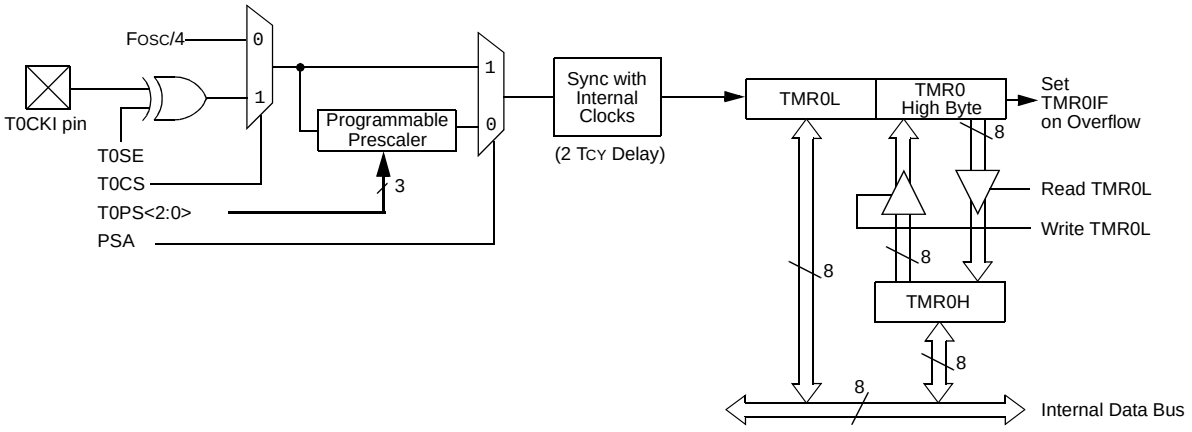
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **WPUB<7:0>:** Weak Pull-up Register bits

1 = Pull-up enabled on PORT pin

0 = Pull-up disabled on PORT pin

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

11.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable; its value is set by the PSA and T0PS<2:0> bits of the T0CON register which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When the prescaler is assigned, prescale values from 1:2 through 1:256 in integer power-of-2 increments are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) clear the prescaler count.

Note:	Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

11.4.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed “on-the-fly” during program execution.

11.5 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit of the INTCON register. Before re-enabling the interrupt, the TMR0IF bit must be cleared by software in the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 11-1: REGISTERS ASSOCIATED WITH TIMER0

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	109
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	110
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS<2:0>			154
TMR0H	Timer0 Register, High Byte								—
TMR0L	Timer0 Register, Low Byte								—
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151

Legend: — = unimplemented locations, read as ‘0’. Shaded bits are not used by Timer0.

TABLE 14-10: REGISTERS ASSOCIATED WITH STANDARD PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				198
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				198
CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<3:0>				198
CCP4CON	—	—	DC4B<1:0>		CCP4M<3:0>				198
CCP5CON	—	—	DC5B<1:0>		CCP5M<3:0>				198
CCPTMRS0	C3TSEL<1:0>		—	C2TSEL<1:0>		—	C1TSEL<1:0>		201
CCPTMRS1	—	—	—	—	C5TSEL<1:0>		C4TSEL<1:0>		201
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	122
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP	124
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	118
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE	120
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	113
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF	115
PMD0	UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	52
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
PR2	Timer2 Period Register								—
PR4	Timer4 Period Register								—
PR6	Timer6 Period Register								—
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		166
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		166
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		166
TMR2	Timer2 Register								—
TMR4	Timer4 Register								—
TMR6	Timer6 Register								—
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	151
TRISE	WPUE3	—	—	—	—	TRISE2(1)	TRISE1(1)	TRISE0(1)	151

Legend: — = Unimplemented location, read as '0'. Shaded bits are not used by Standard PWM mode.

Note 1: These registers/bits are available on PIC18(L)F4XK22 devices.

TABLE 14-11: CONFIGURATION REGISTERS ASSOCIATED WITH STANDARD PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	348

Legend: — = Unimplemented location, read as '0'. Shaded bits are not used by Standard PWM mode.

REGISTER 14-5: ECCPxAS: CCPX AUTO-SHUTDOWN CONTROL 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
CCPxASE	CCPxAS<2:0>			PSSxAC<1:0>		PSSxBD<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **CCPxASE:** CCPx Auto-shutdown Event Status bit
if PxRSEN = 1;
1 = An Auto-shutdown event occurred; CCPxASE bit will automatically clear when event goes away; CCPx outputs in shutdown state
0 = CCPx outputs are operating
if PxRSEN = 0;
1 = An Auto-shutdown event occurred; bit must be cleared in software to restart PWM; CCPx outputs in shutdown state
0 = CCPx outputs are operating

bit 6-4 **CCPxAS<2:0>:** CCPx Auto-Shutdown Source Select bits **(1)**
000 = Auto-shutdown is disabled
001 = Comparator C1 (async_C1OUT) – output high will cause shutdown event
010 = Comparator C2 (async_C2OUT) – output high will cause shutdown event
011 = Either Comparator C1 or C2 – output high will cause shutdown event
100 = FLT0 pin – low level will cause shutdown event
101 = FLT0 pin – low level or Comparator C1 (async_C1OUT) – high level will cause shutdown event
110 = FLT0 pin – low level or Comparator C2 (async_C2OUT) – high level will cause shutdown event
111 = FLT0 pin – low level or Comparators C1 or C2 – high level will cause shutdown event

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

Note 1: If C1SYNC or C2SYNC bits in the CM2CON1 register are enabled, the shutdown will be delayed by Timer1.

15.2.6 SPI OPERATION IN SLEEP MODE

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

Special care must be taken by the user when the MSSPx clock is much faster than the system clock.

In Slave mode, when MSSPx interrupts are enabled, after the master completes sending data, an MSSPx interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSPx interrupts should be disabled.

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

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

TABLE 15-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	149
ANSELB	—	—	ANSB5	ANSB4	ANSB3(1)	ANSB2(1)	ANSB1(1)	ANSB0(1)	150
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	150
ANSELD	ANSD7	ANSD6	ANSD5	ANSD4(2)	ANSD3(2)	ANSD2	ANSD1(2)	ANSD0(2)	150
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
SSP1BUF	SSP1 Receive Buffer/Transmit Register								—
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				253
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	256
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	252
SSP2BUF	SSP2 Receive Buffer/Transmit Register								—
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				253
SSP2CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	256
SSP2STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	252
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3(1)	TRISB2(1)	TRISB1(1)	TRISB0(1)	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD	TRISD7	TRISD6	TRISD5	TRISD4(2)	TRISD3(2)	TRISD2	TRISD1(2)	TRISD0(2)	151

Legend: Shaded bits are not used by the MSSPx in SPI mode.

Note 1: PIC18(L)F2XK22 devices.

2: PIC18(L)F4XK22 devices.

18.9 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR x CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
CxON	CxOUT	CxOE	CxPOL	CxSP	CxR	CxCH<1:0>	
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 **CxON:** Comparator Cx Enable bit
1 = Comparator Cx is enabled
0 = Comparator Cx is disabled

bit 6 **CxOUT:** Comparator Cx Output bit
If CxPOL = 1 (inverted polarity):
CxOUT = 0 when CxVIN+ > CxVIN-
CxOUT = 1 when CxVIN+ < CxVIN-
If CxPOL = 0 (non-inverted polarity):
CxOUT = 1 when CxVIN+ > CxVIN-
CxOUT = 0 when CxVIN+ < CxVIN-

bit 5 **CxOE:** Comparator Cx Output Enable bit
1 = CxOUT is present on the CxOUT pin(1)
0 = CxOUT is internal only

bit 4 **CxPOL:** Comparator Cx Output Polarity Select bit
1 = CxOUT logic is inverted
0 = CxOUT logic is not inverted

bit 3 **CxSP:** Comparator Cx Speed/Power Select bit
1 = Cx operates in Normal-Power, Higher Speed mode
0 = Cx operates in Low-Power, Low-Speed mode

bit 2 **CxR:** Comparator Cx Reference Select bit (non-inverting input)
1 = CxVIN+ connects to CxVREF output
0 = CxVIN+ connects to C12IN+ pin

bit 1-0 **CxCH<1:0>:** Comparator Cx Channel Select bit
00 = C12IN0- pin of Cx connects to CxVIN-
01 = C12IN1- pin of Cx connects to CxVIN-
10 = C12IN2- pin of Cx connects to CxVIN-
11 = C12IN3- pin of Cx connects to CxVIN-

Note 1: Comparator output requires the following three conditions: CxOE = 1, CxON = 1 and corresponding port TRIS bit = 0.

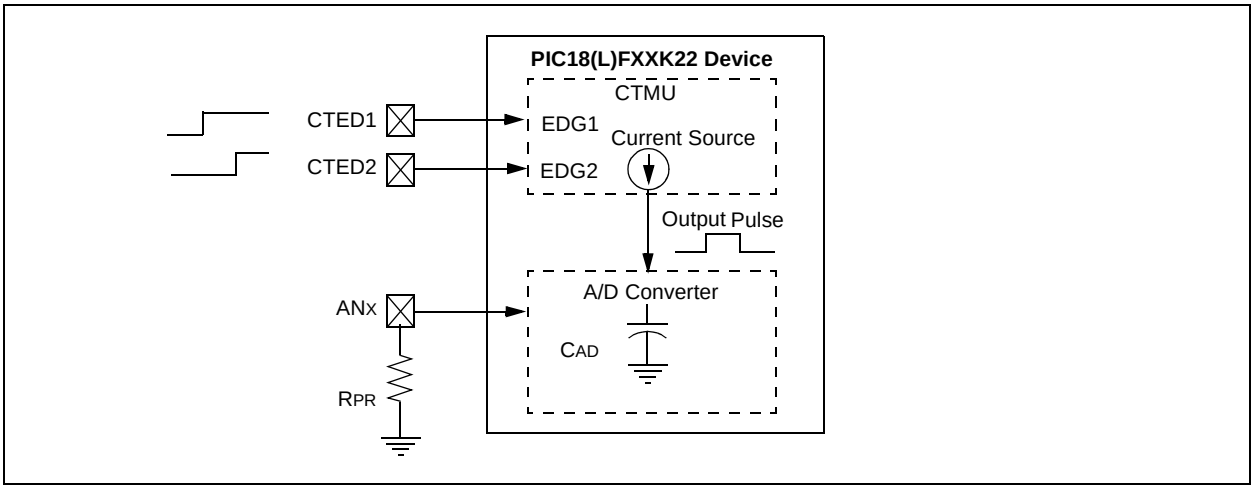
19.5 Measuring Time with the CTMU Module

Time can be precisely measured after the ratio (C/I) is measured from the current and capacitance calibration step by following these steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT.
3. Set EDG2STAT.
4. Perform an A/D conversion.
5. Calculate the time between edges as $T = (C/I) * V$, where I is calculated in the current calibration step (**Section 19.3.1 "Current Source Calibration"**), C is calculated in the capacitance calibration step (**Section 19.3.2 "Capacitance Calibration"**) and V is measured by performing the A/D conversion.

It is assumed that the time measured is small enough that the capacitance, C_{OFFSET}, provides a valid voltage to the A/D Converter. For the smallest time measurement, always set the A/D Channel Select register (AD1CHS) to an unused A/D channel; the corresponding pin for which is not connected to any circuit board trace. This minimizes added stray capacitance, keeping the total circuit capacitance close to that of the A/D Converter itself (4-5 pF). To measure longer time intervals, an external capacitor may be connected to an A/D channel and this channel selected when making a time measurement.

FIGURE 19-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT



19.9 Effects of a Reset on CTMU

Upon Reset, all registers of the CTMU are cleared. This leaves the CTMU module disabled, its current source is turned off and all configuration options return to their default settings. The module needs to be re-initialized following any Reset.

If the CTMU is in the process of taking a measurement at the time of Reset, the measurement will be lost. A partial charge may exist on the circuit that was being measured, and should be properly discharged before the CTMU makes subsequent attempts to make a measurement. The circuit is discharged by setting and then clearing the IDISSEN bit (CTMUCONH<1>) while the A/D Converter is connected to the appropriate channel.

19.10 Registers

There are three control registers for the CTMU:

- CTMUCONH
- CTMUCONL
- CTMUICON

The CTMUCONH and CTMUCONL registers (Register 19-1 and Register 19-2) contain control bits for configuring the CTMU module edge source selection, edge source polarity selection, edge sequencing, A/D trigger, analog circuit capacitor discharge and enables. The CTMUICON register (Register 19-3) has bits for selecting the current source range and current source trim.

19.11 Register Definitions: CTMU Control

REGISTER 19-1: CTMUCONH: CTMU CONTROL REGISTER 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 4 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 3 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 2 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 1 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 0 **CTTRIG:** CTMU Special Event Trigger Control Bit
1 = CTMU Special Event Trigger is enabled
0 = CTMU Special Event Trigger is disabled

TABLE 24-1: CONFIGURATION BITS AND DEVICE IDs

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	—	—	—	—	—	—	—	—	0000 0000
300001h	CONFIG1H	IESO	FCMEN	PRICLKEN	PLLCFG	FOSC<3:0>				0010 0101
300002h	CONFIG2L	—	—	—	BORV<1:0>		BOREN<1:0>		$\overline{\text{PWRTEN}}$	0001 1111
300003h	CONFIG2H	—	—	WDPS<3:0>				WDTEN<1:0>		0011 1111
300004h	CONFIG3L	—	—	—	—	—	—	—	—	0000 0000
300005h	CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	1011 1111
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	XINST	—	—	—	LVP(1)	—	STRVEN	1000 0101
300007h	CONFIG4H	—	—	—	—	—	—	—	—	1111 1111
300008h	CONFIG5L	—	—	—	—	CP3(2)	CP2(2)	CP1	CP0	0000 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	1100 0000
30000Ah	CONFIG6L	—	—	—	—	WRT3(2)	WRT2(2)	WRT1	WRT0	0000 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC(3)	—	—	—	—	—	1110 0000
30000Ch	CONFIG7L	—	—	—	—	EBTR3(2)	EBTR2(2)	EBTR1	EBTR0	0000 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	0100 0000
3FFFFEh	DEVID1(4)	DEV<2:0>			REV<4:0>					qqqq qqqq
3FFFFFh	DEVID2(4)	DEV<10:3>								0101 qqqq

Legend: – = unimplemented, q = value depends on condition. Shaded bits are unimplemented, read as '0'.

Note 1: Can only be changed when in high voltage programming mode.

2: Available on PIC18(L)FX5K22 and PIC18(L)FX6K22 devices only.

3: In user mode, this bit is read-only and cannot be self-programmed.

4: See Register 24-12 and Register 24-13 for DEVID values. DEVID registers are read-only and cannot be programmed by the user.

XORWF **Exclusive OR W with f**

Syntax: XORWF f {,d {,a}}

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

Operation: (W) .XOR. (f) → dest

Status Affected: N, Z

Encoding:

0001	10da	ffff	ffff

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).
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 $f \le 95$ (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 to destination

Example: `XORWF REG, 1, 0`

Before Instruction
REG = AFh
W = B5h
After Instruction
REG = 1Ah
W = B5h

27.5 DC Characteristics: Primary Run Supply Current, PIC18(L)F2X/4XK22

PIC18LF2X/4XK22		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C					
PIC18F2X/4XK22		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Device Characteristics**	**Typ**	**Max**	**Units**	**Conditions**		
D070	Supply Current (IDD)[1],[2]	0.11	0.20	mA	-40°C to +125°C	VDD = 1.8V	FOSC = 1 MHz (**PRI_RUN** mode, ECM source)
D071		0.17	0.25	mA	-40°C to +125°C	VDD = 3.0V	
D072		0.15	0.25	mA	-40°C to +125°C	VDD = 2.3V	FOSC = 1 MHz (**PRI_RUN** mode, ECM source)
D073		0.20	0.30	mA	-40°C to +125°C	VDD = 3.0V	
D074		0.25	0.35	mA	-40°C to +125°C	VDD = 5.0V	
D075		1.45	2.0	mA	-40°C to +125°C	VDD = 1.8V	FOSC = 20 MHz (**PRI_RUN** mode, ECH source)
D076		2.60	3.5	mA	-40°C to +125°C	VDD = 3.0V	
D077		1.95	2.5	mA	-40°C to +125°C	VDD = 2.3V	FOSC = 20 MHz (**PRI_RUN** mode, ECH source)
D078		2.65	3.5	mA	-40°C to +125°C	VDD = 3.0V	
D079		2.95	4.5	mA	-40°C to +125°C	VDD = 5.0V	
D080		7.5	10	mA	-40°C to +125°C	VDD = 3.0V	FOSC = 64 MHz (**PRI_RUN**, ECH oscillator)
D081		7.5	10	mA	-40°C to +125°C	VDD = 3.0V	FOSC = 64 MHz (**PRI_RUN** mode, ECH source)
D082		8.5	11.5	mA	-40°C to +125°C	VDD = 5.0V	
D083		1.0	1.5	mA	-40°C to +125°C	VDD = 1.8V	FOSC = 4 MHz 16 MHz Internal (**PRI_RUN** mode, ECM + PLL source)
D084		1.8	3.0	mA	-40°C to +125°C	VDD = 3.0V	
D085		1.4	2.0	mA	-40°C to +125°C	VDD = 2.3V	FOSC = 4 MHz 16 MHz Internal (**PRI_RUN** mode, ECM + PLL source)
D086		1.85	2.5	mA	-40°C to +125°C	VDD = 3.0V	
D087		2.1	3.0	mA	-40°C to +125°C	VDD = 5.0V	
D088		6.35	9.0	mA	-40°C to +125°C	VDD = 3.0V	FOSC = 16 MHz 64 MHz Internal (**PRI_RUN** mode, ECH + PLL source)
D089		6.35	9.0	mA	-40°C to +125°C	VDD = 3.0V	FOSC = 16 MHz 64 MHz Internal (**PRI_RUN** mode, ECH + PLL source)
D090		7.0	10	mA	-40°C to +125°C	VDD = 5.0V	

Note 1: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
Test condition: All Peripheral Module Control bits in PMD0, PMD1 and PMD2 set to '1'.

2: The test conditions for all IDD measurements in active operation mode are:
All I/O pins set as outputs driven to Vss;
MCLR = VDD;
OSC1 = external square wave, from rail-to-rail (PRI_RUN and PRI_IDLE only).

FIGURE 28-62: PIC18F2X/4XK22 TYPICAL IDD: PRI_IDLE EC MEDIUM POWER

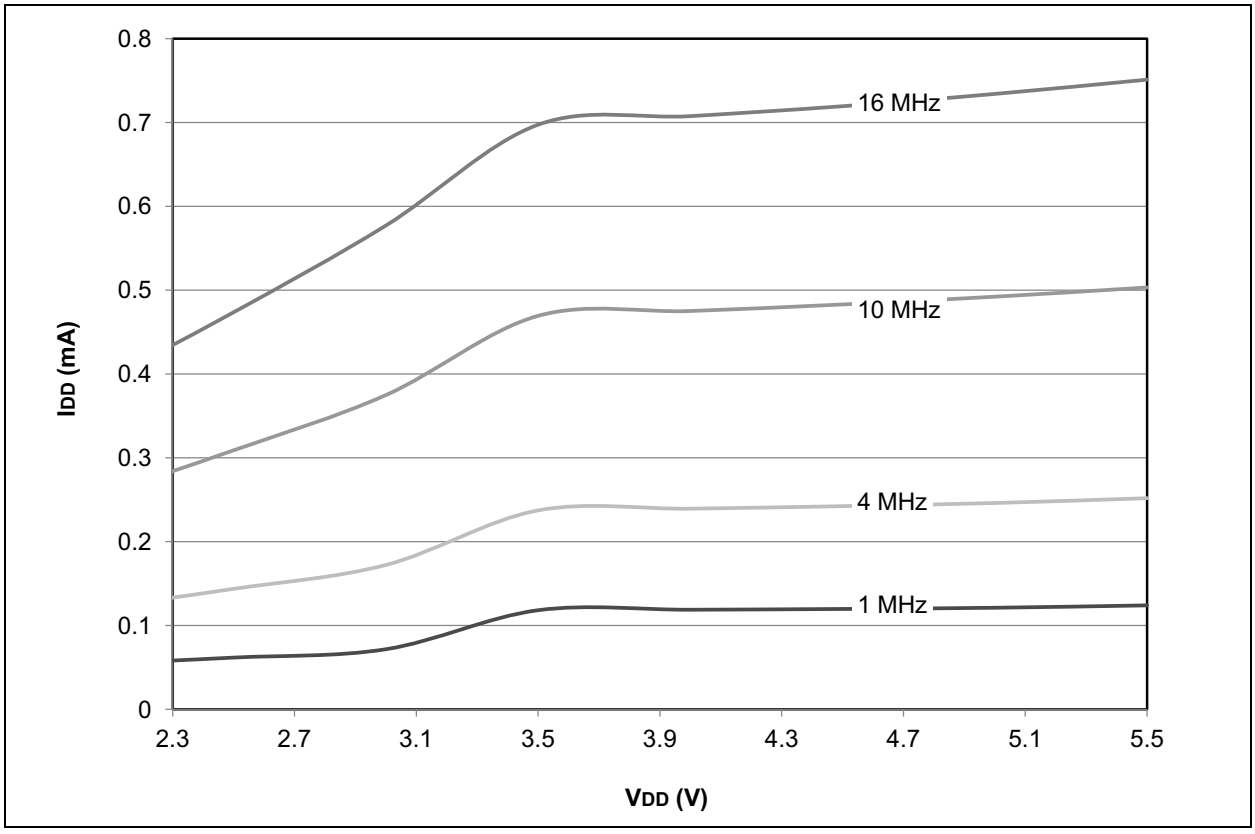


FIGURE 28-63: PIC18F2X/4XK22 MAXIMUM IDD: PRI_IDLE EC MEDIUM POWER