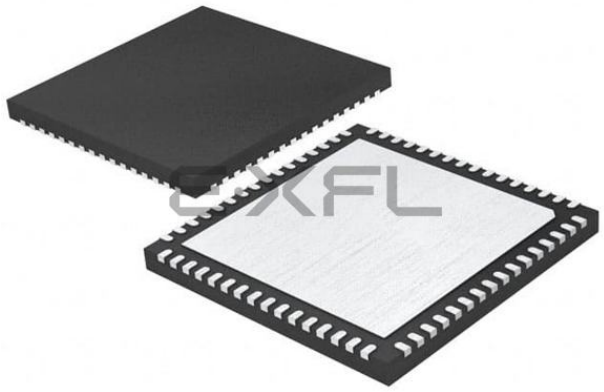


Welcome to E-XFL.COM

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	64KB (32K 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66k22-e-mr

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 VIL, 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.

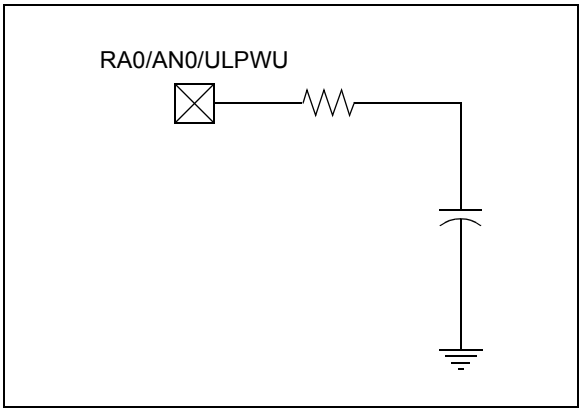
See Example 4-1 for initializing the ULPWU module.

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).

6.4.3.1 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. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that 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. The operands 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.

FIGURE 6-8: INDIRECT ADDRESSING

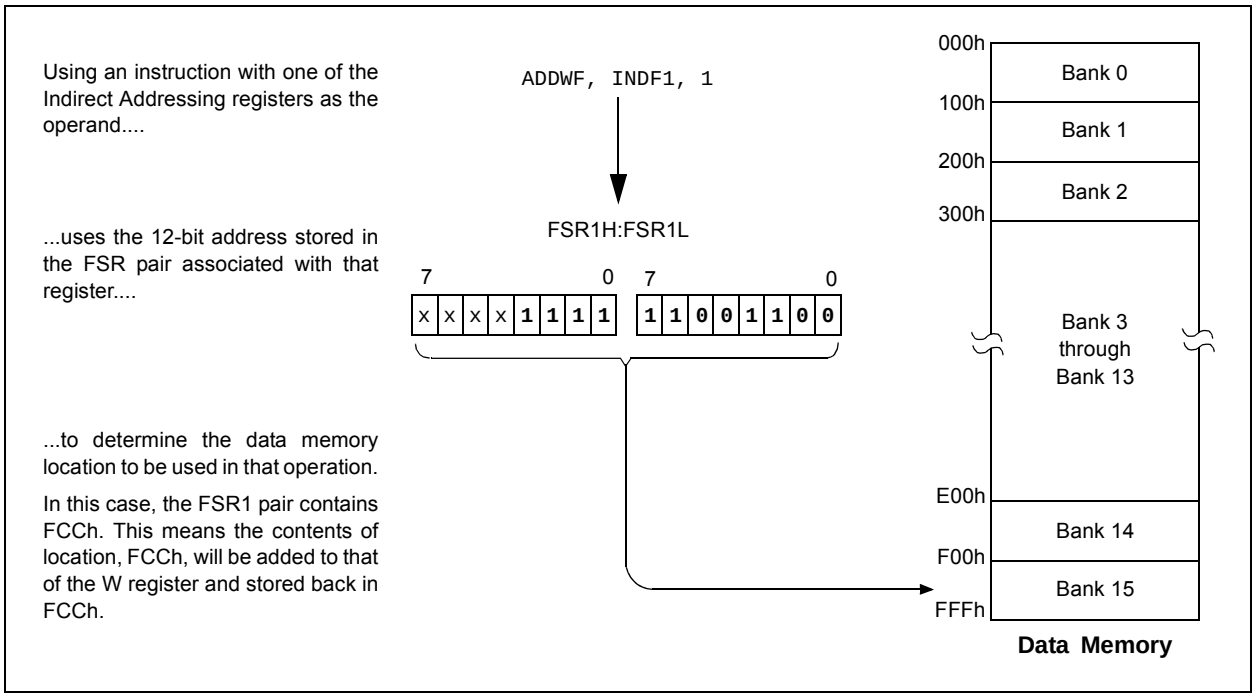
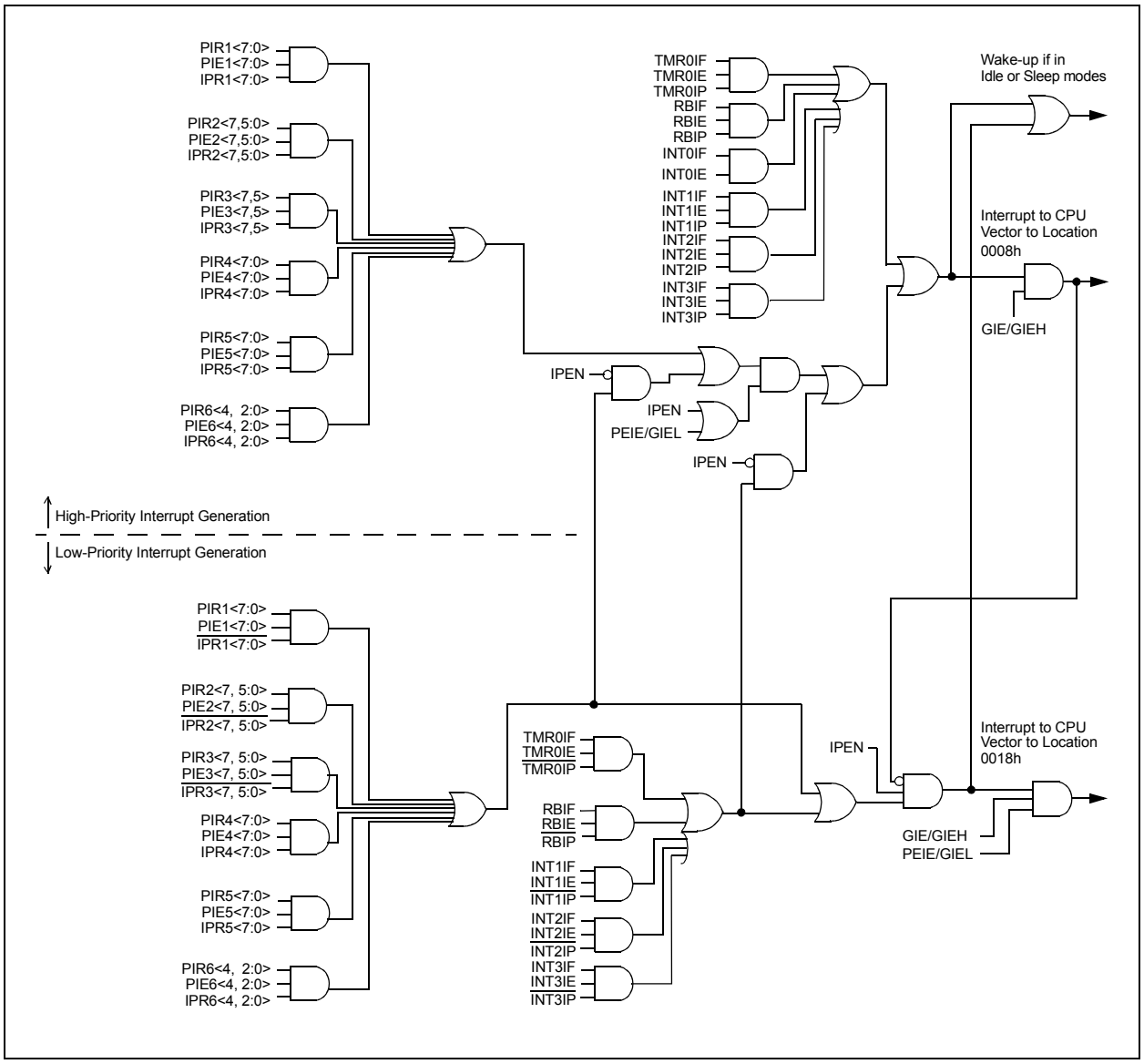


FIGURE 11-1: PIC18F87K22 FAMILY INTERRUPT LOGIC



11.2 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are six Peripheral Interrupt Request (Flag) registers (PIR1 through PIR6).

Note 1: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON<7>).

2: User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

REGISTER 11-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	TMR1GIF	TMR2IF	TMR1IF
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 **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit
1 = A read or write operation has taken place (must be cleared in software)
0 = No read or write operation has occurred

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RC1IF:** EUSART Receive Interrupt Flag bit
1 = The EUSART receive buffer, RCREG1, is full (cleared when RCREG1 is read)
0 = The EUSART receive buffer is empty

bit 4 **TX1IF:** EUSART Transmit Interrupt Flag bit
1 = The EUSART transmit buffer, TXREG1, is empty (cleared when TXREG1 is written)
0 = The EUSART transmit buffer is full

bit 3 **SSP1IF:** Master Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2 **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = Timer gate interrupt occurred (must be cleared in software)
0 = No timer gate interrupt occurred

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

REGISTER 11-6: PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

R/W-0	U-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR5GIF	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF
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 **TMR5GIF:** Timer5 Gate Interrupt Flag bit
1 = Timer gate interrupt occurred (must be cleared in software)
0 = No timer gate interrupt occurred

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

bit 5 **RC2IF:** EUSART Receive Interrupt Flag bit
1 = The EUSART receive buffer, RCREG2, is full (cleared when RCREG2 is read)
0 = The EUSART receive buffer is empty

bit 4 **TX2IF:** EUSART Transmit Interrupt Flag bit
1 = The EUSART transmit buffer, TXREG2, is empty (cleared when TXREG2 is written)
0 = The EUSART transmit buffer is full

bit 3 **CTMUIF:** CTMU Interrupt Flag bit
1 = CTMU interrupt occurred (must be cleared in software)
0 = No CTMU interrupt occurred

bit 2 **CCP2IF:** ECCP2 Interrupt Flag bit
Capture mode:
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred
Compare mode:
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **CCP1IF:** ECCP1 Interrupt Flag bit
Capture mode:
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred
Compare mode:
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred
PWM mode:
Unused in this mode.

bit 0 **RTCCIF:** RTCC Interrupt Flag bit
1 = RTCC interrupt occurred (must be cleared in software)
0 = No RTCC interrupt occurred

NOTES:

FIGURE 14-7: TIMER1 GATE SINGLE PULSE AND TOGGLE COMBINED MODE

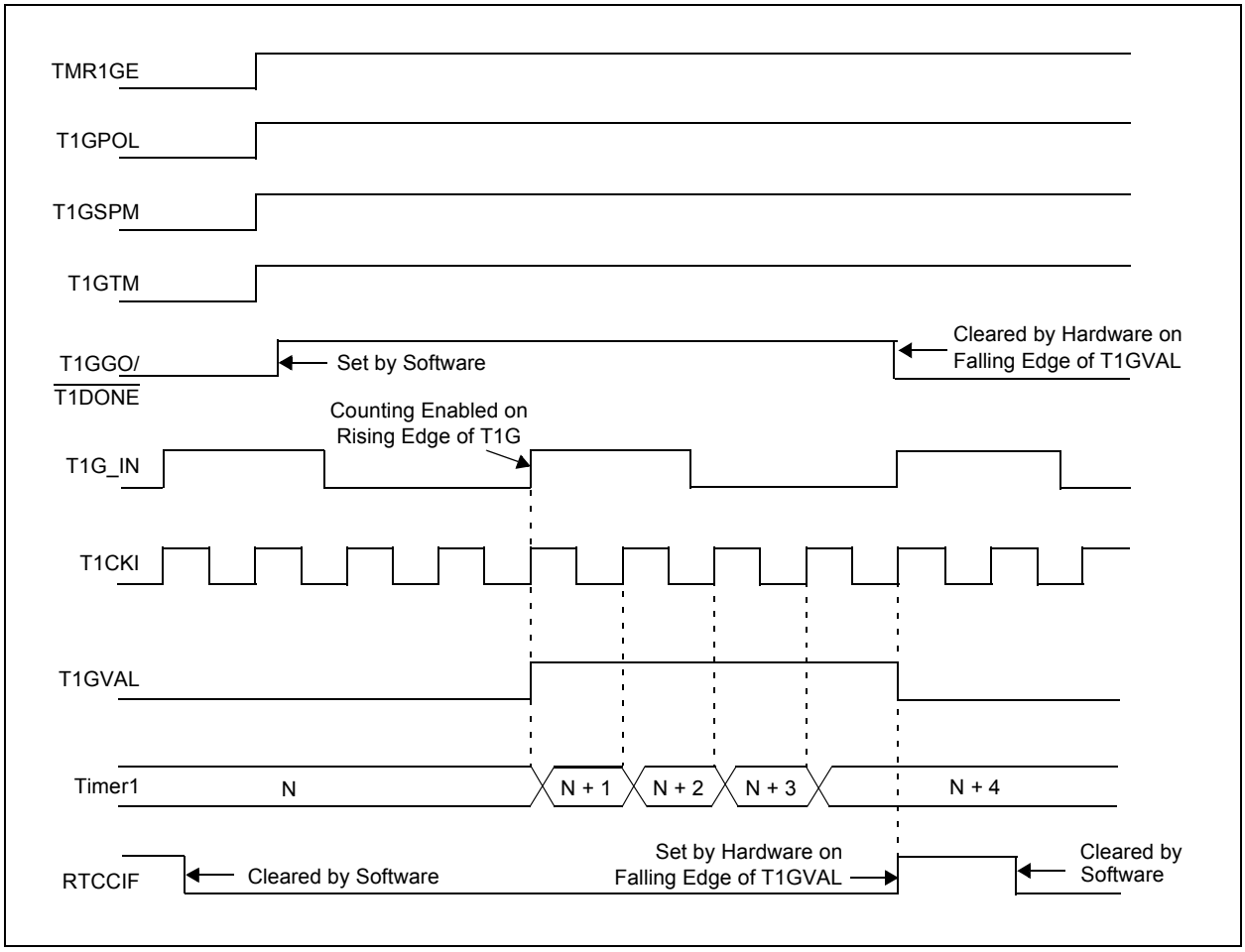


TABLE 14-5: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

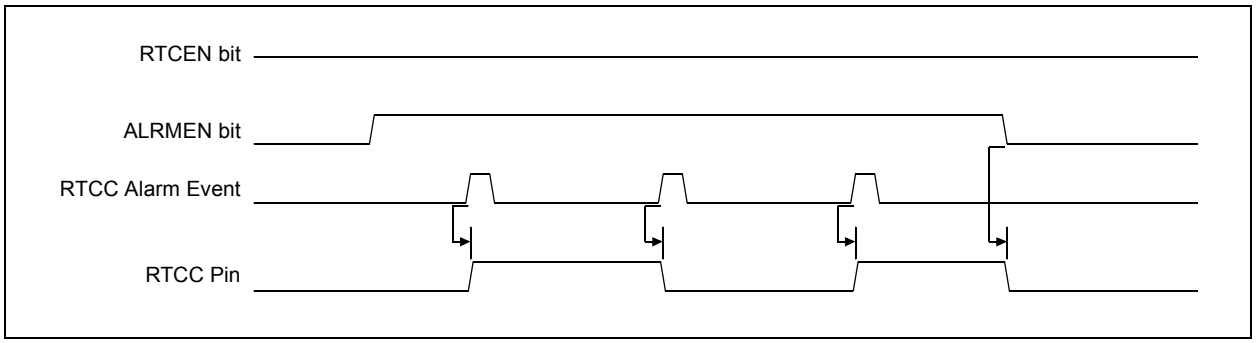
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	TMR1GIF	TMR2IF	TMR1IF
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP
TMR1L	Timer1 Register Low Byte							
TMR1H	Timer1 Register High Byte							
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	SOSCEN	$\overline{\text{T1SYNC}}$	RD16	TMR1ON
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	$\overline{\text{T1GGO/T1DONE}}$	T1GVAL	T1GSS1	T1GSS0
OSCCON2	—	SOSCRUN	—	—	SOSCGO	—	MFIOFS	MFIOSEL
PMD1	PSPMD	CTMUMD	RTCCMD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	EMBDM

Legend: Shaded cells are not used by the Timer1 module.

Note 1: Unimplemented on 32-Kbyte devices (PIC18FX5K22).

NOTES:

FIGURE 18-6: TIMER PULSE GENERATION



18.4 Sleep Mode

The timer and alarm continue to operate while in Sleep mode. The operation of the alarm is not affected by Sleep, as an alarm event can always wake up the CPU.

The Idle mode does not affect the operation of the timer or alarm.

18.5 Reset

18.5.1 DEVICE RESET

When a device Reset occurs, the ALRMRPT register is forced to its Reset state, causing the alarm to be disabled (if enabled prior to the Reset). If the RTCC was enabled, it will continue to operate when a basic device Reset occurs.

18.5.2 POWER-ON RESET (POR)

The RTCCFG and ALRMRPT registers are reset only on a POR. Once the device exits the POR state, the clock registers should be reloaded with the desired values.

The timer prescaler values can be reset only by writing to the SECONDS register. No device Reset can affect the prescalers.

NOTES:

REGISTER 20-1: CCPxCON: ENHANCED CAPTURE/COMPARE/PWMx CONTROL

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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 **PxM<1:0>:** Enhanced PWM Output Configuration bits

If CCPxM<3:2> = 00, 01, 10:

xx = PxA is assigned as capture/compare input/output; PxB, PxC and PxD are assigned as port pins

If CCPxM<3:2> = 11:

00 = Single output: PxA, PxB, PxC and PxD are controlled by steering (see **Section 20.4.7 "Pulse Steering Mode"**)

01 = Full-bridge output forward: PxD is modulated; PxA is active; PxB, PxC are inactive

10 = Half-bridge output: PxA, PxB are modulated with dead-band control; PxC and PxD are assigned as port pins

11 = Full-bridge output reverse: PxB is modulated; PxC is active; PxA and PxD are inactive

bit 5-4 **DCxB<1:0>**: PWM Duty Cycle Bit 1 and Bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM<3:0>**: ECCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCPx module)

0001 = Reserved

0010 = Compare mode: toggle output on match

0011 = Capture mode

0100 = Capture mode: every falling edge

0101 = Capture mode: every rising edge

0110 = Capture mode: every fourth rising edge

0111 = Capture mode: every 16th rising edge

1000 = Compare mode: initialize ECCPx pin low, set output on compare match (set CCPxIF)

1001 = Compare mode: initialize ECCPx pin high, clear output on compare match (set CCPxIF)

1010 = Compare mode: generate software interrupt only, ECCPx pin reverts to I/O state

1011 = Compare mode: trigger special event (ECCPx resets TMR1 or TMR3, starts A/D conversion, sets CCPxIF bit)

1100 = PWM mode: PxA and PxC are active-high; PxB and PxD are active-high

1101 = PWM mode: PxA and PxC are active-high; PxB and PxD are active-low

1110 = PWM mode: PxA and PxC are active-low; PxB and PxD are active-high

1111 = PWM mode: PxA and PxC are active-low; PxB and PxD are active-low

21.4.6.1 I²C™ Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDAx while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted, 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

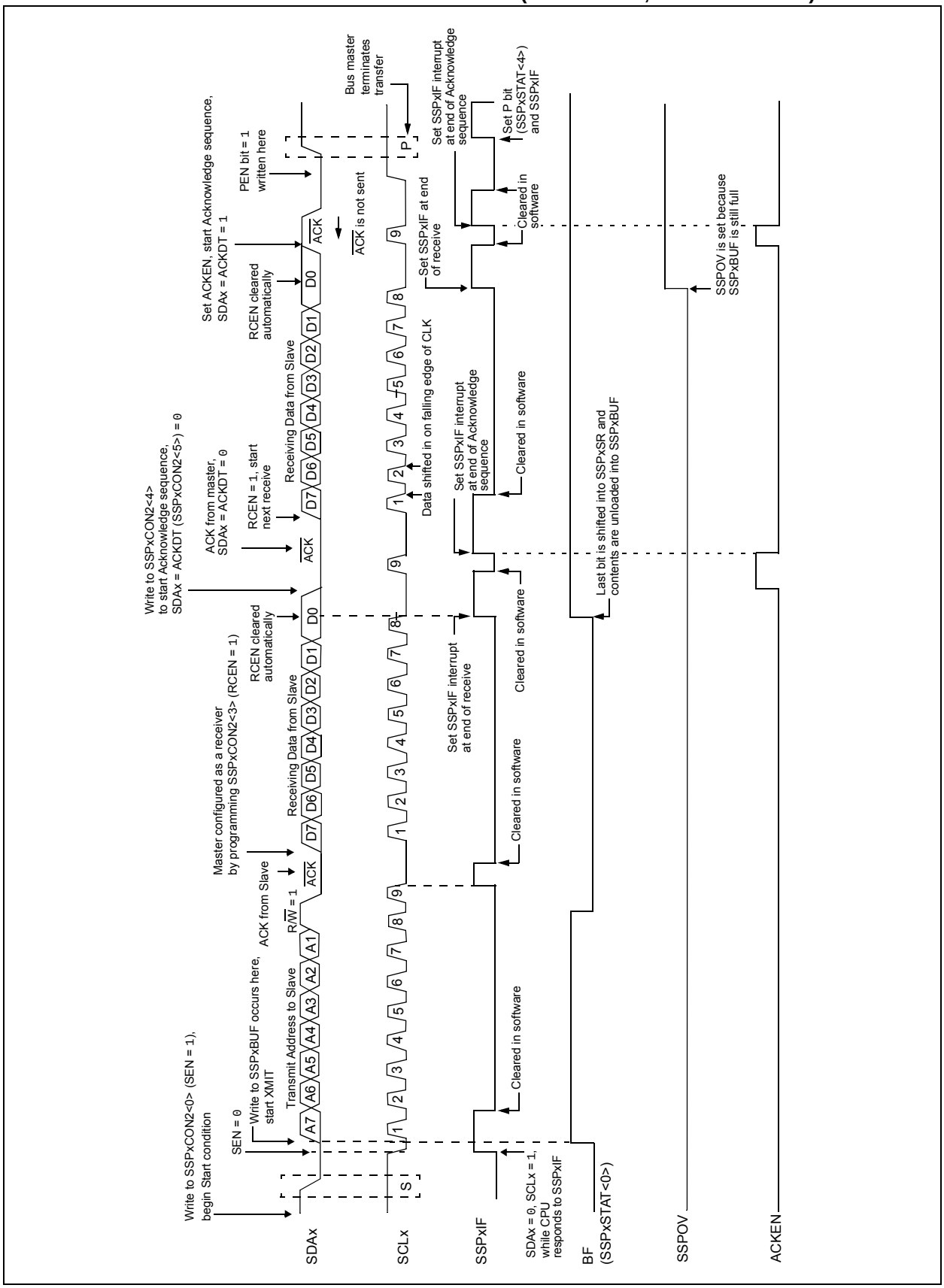
In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address, followed by a '1' to indicate the receive bit. Serial data is received via SDAx, while SCLx outputs the serial clock. Serial data is received, 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator, used for the SPI mode operation, is used to set the SCLx clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 21.4.7 "Baud Rate"** for more details.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPxCON2<0>).
2. SSPxIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPxBUF with the slave address to transmit.
4. Address is shifted out the SDAx pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPxCON2 register (SSPxCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
7. The user loads the SSPxBUF with eight bits of data.
8. Data is shifted out the SDAx pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPxCON2 register (SSPxCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPxCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

FIGURE 21-24: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



REGISTER 23-10: ANCON2: A/D PORT CONFIGURATION REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSEL23(1)	ANSEL22(1)	ANSEL21(1)	ANSEL20(1)	ANSEL19	ANSEL18	ANSEL17	ANSEL16
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 **ANSEL<23:16>:** Analog Port Configuration bits (AN23 through AN16)(1)

1 = Pin is configured as an analog channel; digital input is disabled and any inputs read as '0'
0 = Pin is configured as a digital port

Note 1: AN15 through AN12 and AN23 through AN20 are implemented only on 80-pin devices. For 64-pin devices, the corresponding ANSELx bits are still implemented for these channels, but have no effect.

The analog reference voltage is software-selectable to either the device's positive and negative supply voltage (AV_{DD} and AV_{SS}) or the voltage level on the RA3/AN3/V_{REF}+ and RA2/AN2/V_{REF}- pins. V_{REF}+ has two additional Internal Reference Voltage selections: 2.048V and 4.096V.

The A/D Converter can uniquely operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D Converter's internal RC oscillator.

The output of the Sample-and-Hold (S/H) is the input into the converter, which generates the result via successive approximation.

Each port pin associated with the A/D Converter can be configured as an analog input or a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>) is cleared and the A/D Interrupt Flag bit, ADIF (PIR1<6>), is set.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted. The value in the ADRESH:ADRESL register pair is not modified for a Power-on Reset. These registers will contain unknown data after a Power-on Reset.

The block diagram of the A/D module is shown in Figure 23-4.

24.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake up the device from Sleep mode, when enabled. Each operational comparator will consume additional current.

To minimize power consumption while in Sleep mode, turn off the comparators (CON = 0) before entering Sleep. If the device wakes up from Sleep, the contents of the CMxCON register are not affected.

24.8 Effects of a Reset

A device Reset forces the CMxCON registers to their Reset state. This forces both comparators and the voltage reference to the OFF state.

TABLE 24-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR6	—	—	—	EEIF	—	CMP3IF	CMP2IF	CMP1IF
PIE6	—	—	—	EEIE	—	CMP3IE	CMP2IE	CMP1IE
IPR6	—	—	—	EEIP	—	CMP3IP	CMP2IP	CMP1IP
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
CM3CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
CVRCON	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0
CMSTAT	CMP3OUT	CMP2OUT	CMP1OUT	—	—	—	—	—
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—
PORTG	—	—	RG5(1)	RG4	RG3	RG2	RG1	RG0
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0
PORTH	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0
LATH	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0
TRISH	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0
ANCON0	ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
ANCON1	ANSEL15	ANSEL14	ANSEL13	ANSEL12	ANSEL11	ANSEL10	ANSEL9	ANSEL8
ANCON2	ANSEL23	ANSEL22	ANSEL21	ANSEL20	ANSEL19	ANSEL18	ANSEL17	ANSEL16
PMD0	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSP2MD	SSP1MD	ADCMD

Legend: — = unimplemented, read as '0'.

Note 1: Bit is available when Master Clear is disabled (MCLRE = 0). When MCLRE is set, the bit is unimplemented.

27.0 CHARGE TIME MEASUREMENT UNIT (CTMU)

The Charge Time Measurement Unit (CTMU) is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. By working with other on-chip analog modules, the CTMU can precisely measure time, capacitance and relative changes in capacitance or generate output pulses with a specific time delay. The CTMU is ideal for interfacing with capacitive-based sensors.

The module includes these key features:

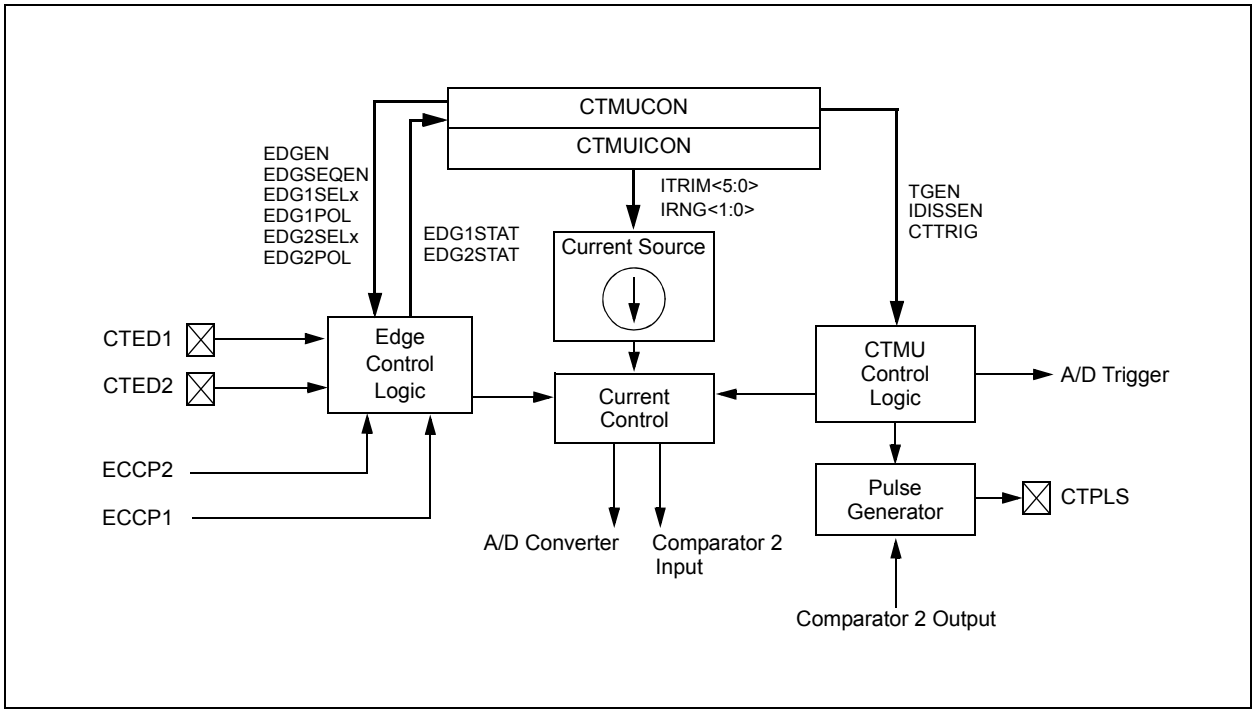
- Up to 24 channels available for capacitive or time measurement input
- On-chip precision current source
- Four-edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edges
- Time measurement resolution of 1 nanosecond
- High-precision time measurement
- Time delay of external or internal signal asynchronous to system clock
- Accurate current source suitable for capacitive measurement

The CTMU works in conjunction with the A/D Converter to provide up to 24 channels for time or charge measurement, depending on the specific device and the number of A/D channels available. When configured for time delay, the CTMU is connected to one of the analog comparators. The level-sensitive input edge sources can be selected from four sources: two external inputs or the ECCP1/ECCP2 Special Event Triggers.

The CTMU special event can trigger the Analog-to-Digital Converter module.

Figure 27-1 provides a block diagram of the CTMU.

FIGURE 27-1: CTMU BLOCK DIAGRAM



REGISTER 28-8: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

R/C-1	R/C-1	R/C-1	R/C-1	R/C-1	R/C-1	R/C-1	R/C-1
CP7(1)	CP6(1)	CP5(1)	CP4(1)	CP3	CP2	CP1	CP0
bit 7							bit 0

Legend:	C = Clearable bit		
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 **CP7:** Code Protection bit(1)
1 = Block 7 is not code-protected(2)
0 = Block 7 is code-protected(2)

bit 6 **CP6:** Code Protection bit(1)
1 = Block 6 is not code-protected(2)
0 = Block 6 is code-protected(2)

bit 5 **CP5:** Code Protection bit(1)
1 = Block 5 is not code-protected(2)
0 = Block 5 is code-protected(2)

bit 4 **CP4:** Code Protection bit(1)
1 = Block 4 is not code-protected(2)
0 = Block 4 is code-protected(2)

bit 3 **CP3:** Code Protection bit
1 = Block 3 is not code-protected(2)
0 = Block 3 is code-protected(2)

bit 2 **CP2:** Code Protection bit
1 = Block 2 is not code-protected(2)
0 = Block 2 is code-protected(2)

bit 1 **CP1:** Code Protection bit
1 = Block 1 is not code-protected(2)
0 = Block 1 is code-protected(2)

bit 0 **CP0:** Code Protection bit
1 = Block 0 is not code-protected(2)
0 = Block 0 is code-protected(2)

Note 1: This bit is available only on PIC18F67K22 and PIC18F87K22 devices.
2: For the memory size of the blocks, see Figure 28-6.

REGISTER 28-14: DEVID1: DEVICE ID REGISTER 1 FOR THE PIC18F87K22 FAMILY

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
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 **DEV<2:0>:** Device ID bits

Devices with DEV<10:3> of '0101 0010' (see DEVID2):
010 = PIC18F65K22
000 = PIC18F66K22
101 = PIC18F85K22
011 = PIC18F86K22

Devices with DEV<10:3> of '0101 0001':
000 = PIC18F67K22
010 = PIC18F87K22

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to indicate the device revision.

REGISTER 28-15: DEVID2: DEVICE ID REGISTER 2 FOR THE PIC18F87K22 FAMILY

R	R	R	R	R	R	R	R
DEV10(1)	DEV9(1)	DEV8(1)	DEV7(1)	DEV6(1)	DEV5(1)	DEV4(1)	DEV3(1)
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 **DEV<10:3>:** Device ID bits(1)

These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.
0101 0010 = PIC18F65K22, PIC18F66K22, PIC18F85K22 and PIC18F86K22
0101 0001 = PIC18F67K22 and PIC18F87K22

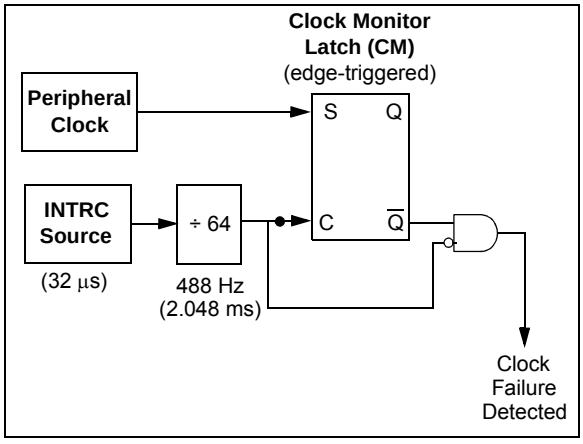
Note 1: These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

28.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation in the event of an external oscillator failure by automatically switching the device clock to the internal oscillator block. The FSCM function is enabled by setting the FCMEN Configuration bit.

When FSCM is enabled, the LF-INTOSC oscillator runs at all times to monitor clocks to peripherals and provide a backup clock in the event of a clock failure. Clock monitoring (shown in Figure 28-4) is accomplished by creating a sample clock signal, which is the output from the LF-INTOSC, divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral device clock and the sample clock are presented as inputs to the Clock Monitor (CM) latch. The CM is set on the falling edge of the device clock source, but cleared on the rising edge of the sample clock.

FIGURE 28-4: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 28-5). This causes the following:

- The FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>)
- The device clock source switches to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the Fail-Safe condition)
- The WDT is reset

During switchover, the postscaler frequency from the internal oscillator block may not be sufficiently stable for timing-sensitive applications. In these cases, it may be desirable to select another clock configuration and enter an alternate power-managed mode. This can be done to attempt a partial recovery or execute a controlled shut-down. See **Section 4.1.4 "Multiple Sleep Commands"** and **Section 28.4.1 "Special Considerations for Using Two-Speed Start-up"** for more details.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF<2:0>, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF<2:0> bits prior to entering Sleep mode.

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

28.5.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTOSC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTOSC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF<2:0> bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed, and decreasing the likelihood of an erroneous time-out.

28.5.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any required start-up delays that are required for the Oscillator mode, such as the OST or PLL timer). The INTOSC multiplexer provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

28.6.2 DATA EEPROM CODE PROTECTION

The entire data EEPROM is protected from external reads and writes by two bits: CPD and WRTD. CPD inhibits external reads and writes of data EEPROM. WRTD inhibits internal and external writes to data EEPROM. The CPU can always read data EEPROM under normal operation, regardless of the protection bit settings.

28.6.3 CONFIGURATION REGISTER PROTECTION

The Configuration registers can be write-protected. The WRTC bit controls protection of the Configuration registers. In normal Execution mode, the WRTC bit is readable only. WRTC can only be written via ICSP or an external programmer.

28.7 ID Locations

Eight memory locations (200000h-200007h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are both readable and writable, during normal execution, through the TBLRD and TBLWT instructions or during program/verify. The ID locations can be read when the device is code-protected.

28.8 In-Circuit Serial Programming

The PIC18F87K22 family of devices can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

For the various Programming modes, see the device programming specification.

28.9 In-Circuit Debugger

When the $\overline{\text{DEBUG}}$ Configuration bit is programmed to a '0', the In-Circuit Debugger (ICD) functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use. Table 28-5 shows which resources are required by the background debugger.

TABLE 28-5: DEBUGGER RESOURCES

I/O Pins:	RB6, RB7
Stack:	Two levels
Program Memory:	512 bytes
Data Memory:	10 bytes

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to $\overline{\text{MCLR}}$/RG5/V_{PP}, V_{DD}, V_{SS}, RB7 and RB6. This will interface to the In-Circuit Debugger module, available from Microchip or one of the third-party development tool companies.