

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](#)"

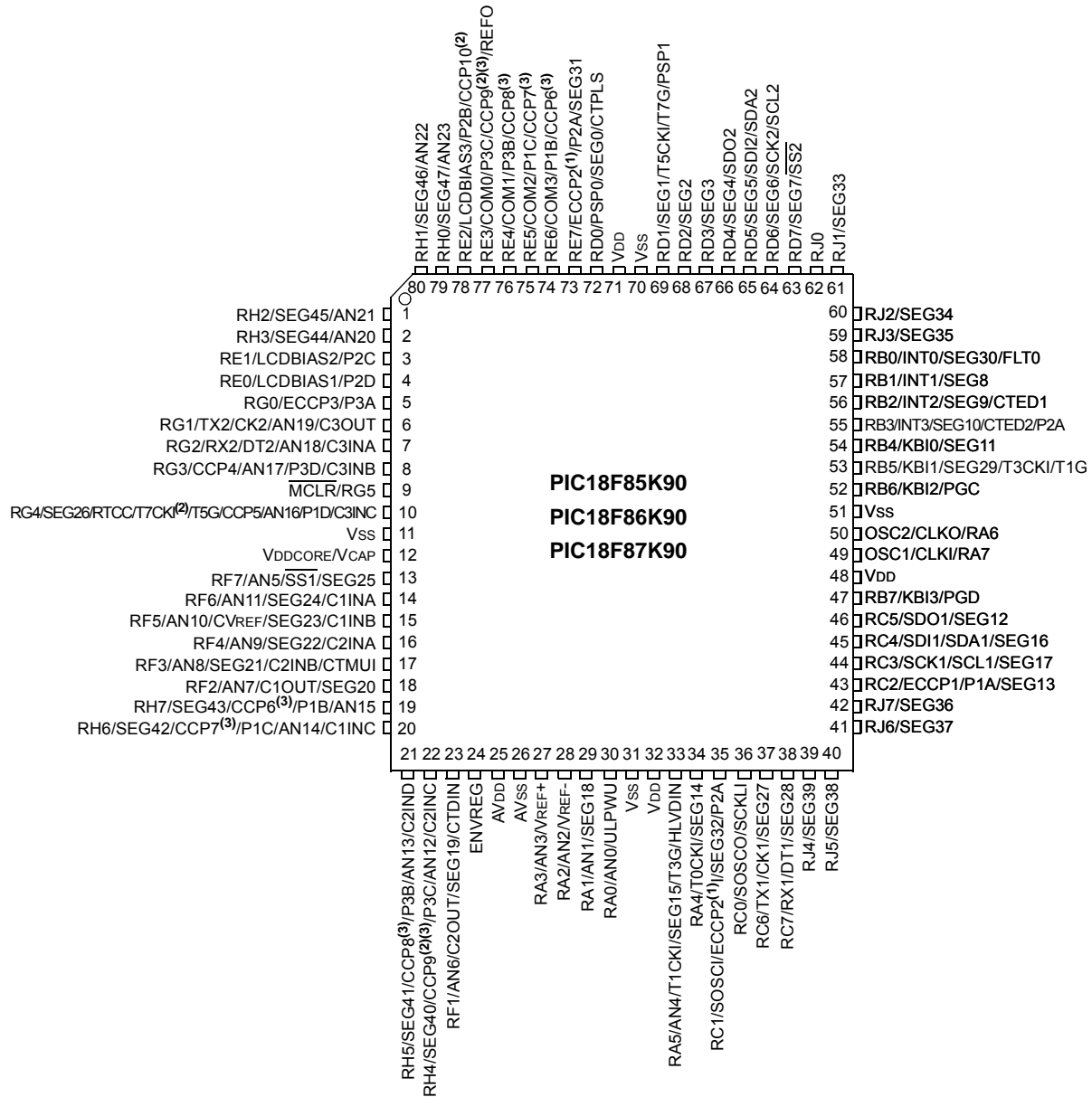
Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, 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 ~ 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/pic18f66k90-i-ptsl

PIC18F87K90 FAMILY

Pin Diagrams – PIC18F8XK90

80-Pin TQFP



- Note** 1: The ECCP2 pin placement depends on the CCP2MX Configuration bit setting.
2: Not available on the PIC18F65K90 and PIC18F85K90.
3: The CCP6, CCP7, CCP8 and CCP9 pin placement depends on the ECCPMX Configuration bit setting

PIC18F87K90 FAMILY

TABLE 1-4: PIC18F8XK90 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
RG0/ECCP3/P3A	5			PORTG is a bidirectional I/O port.
RG0		I/O	ST	Digital I/O.
ECCP3		I/O	ST	Capture 3 input/Compare 3 output/PWM3 output.
P3A		O	—	ECCP3 PWM Output A.
RG1/TX2/CK2/AN19/C3OUT	6			
RG1		I/O	ST	Digital I/O.
TX2		O	—	EUSART asynchronous transmit.
CK2		I/O	ST	EUSART synchronous clock (see related RX2/DT2).
AN19		I	Analog	Analog Input 19.
C3OUT		O	—	Comparator 3 output.
RG2/RX2/DT2/AN18/C3INA	7			
RG2		I/O	ST	Digital I/O.
RX2		I	ST	EUSART asynchronous receive.
DT2		I/O	ST	EUSART synchronous data (see related TX2/CK2).
AN18		I	Analog	Analog Input 18.
C3INA		I	Analog	Comparator 3 Input A.
RG3/CCP4/AN17/P3D/C3INB	8			
RG3		I/O	ST	Digital I/O.
CCP4		I/O	ST	Capture 4 input/Compare 4 output/PWM4 output.
AN17		I	Analog	Analog Input 17.
P3D		O	—	ECCP3 PWM Output D.
C3INB		I	Analog	Comparator 3 Input B.
RG4/SEG26/RTCC/T7CKI/T5G/CCP5/AN16/P1D/C3INC	10			
RG4		I/O	ST	Digital I/O.
SEG26		O	Analog	SEG26 output for LCD.
RTCC		O	—	RTCC output.
T7CKI ⁽³⁾		I	ST	Timer7 clock input.
T5G		I	ST	Timer5 external clock gate input.
CCP5		I/O	ST	Capture 5 input/Compare 5 output/PWM5 output.
AN16		I	Analog	Analog Input 16.
P1D		O	—	ECCP1 PWM Output D.
C3INC		I	Analog	Comparator 3 Input C.
RG5	9			See the $\overline{\text{MCLR}}$ /RG5 pin.

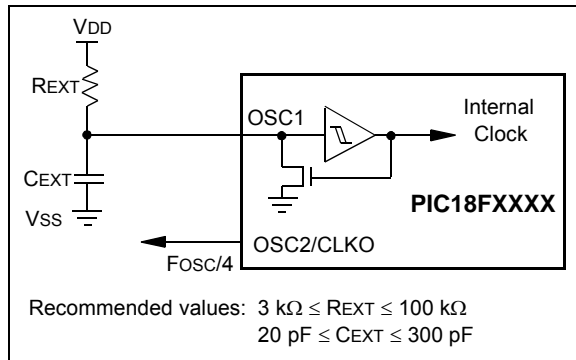
Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels Analog = Analog input
I = Input O = Output
P = Power OD = Open-Drain (no P diode to V_{DD})
I²C™ = I²C/SMBus

- Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.
Note 2: Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.
Note 3: Not available on PIC18F65K90 and PIC18F85K90 devices.
Note 4: The CCP6, CCP7, CCP8 and CCP9 pin placement depends on the ECCPMX Configuration bit setting.

PIC18F87K90 FAMILY

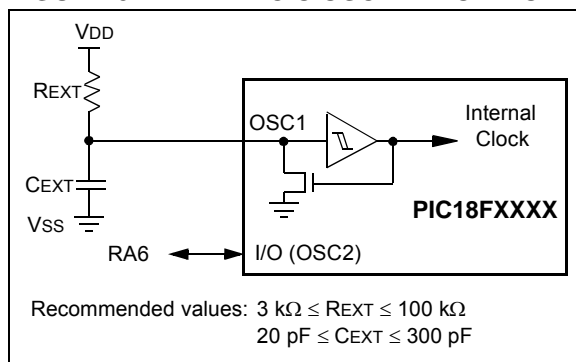
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 3-3 shows how the R/C combination is connected.

FIGURE 3-3: RC OSCILLATOR MODE



The RCIO Oscillator mode (Figure 3-4) functions like the RC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

FIGURE 3-4: RCIO OSCILLATOR MODE

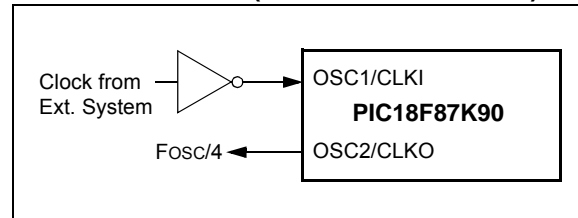


3.5.1 EXTERNAL CLOCK INPUT (EC MODES)

The EC and ECPLL Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

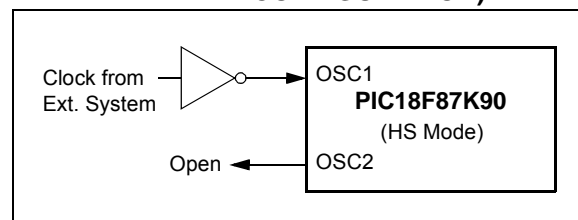
In the EC Oscillator mode, the oscillator frequency, divided by 4, is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 3-5 shows the pin connections for the EC Oscillator mode.

FIGURE 3-5: EXTERNAL CLOCK INPUT OPERATION (EC CONFIGURATION)



An external clock source may also be connected to the OSC1 pin in the HS mode, as shown in Figure 3-6. In this configuration, the divide-by-4 output on OSC2 is not available. Current consumption in this configuration will be somewhat higher than EC mode, as the internal oscillator's feedback circuitry will be enabled (in EC mode, the feedback circuit is disabled).

FIGURE 3-6: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)



3.5.2 PLL FREQUENCY MULTIPLIER

A Phase Lock Loop (PLL) circuit is provided as an option for users who want to use a lower frequency oscillator circuit, or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals, or users who require higher clock speeds from an internal oscillator.

3.5.2.1 HSPLL and ECPLL Modes

The HSPLL and ECPLL modes provide the ability to selectively run the device at four times the external oscillating source to produce frequencies up to 64 MHz.

The PLL is enabled by setting the PLEN bit (OSCTUNE<6>) or the PLLCFG bit (CONFIG1H<4>). The PLEN bit provides software control for the PLL, even if PLLCFG is set to '0'. The PLL is enabled only when the HS or EC oscillator frequency is within the 4 MHz to 16 MHz input range.

This enables additional flexibility for controlling the application's clock speed in software. The PLEN should be enabled in HS or EC Oscillator mode only if the input frequency is in the range of 4 MHz-16 MHz.

PIC18F87K90 FAMILY

3.7 Reference Clock Output

In addition to the FOSC/4 clock output in certain oscillator modes, the device clock in the PIC18F87K90 family can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 3-4). Setting the ROON bit (REFOCON<7>) makes the clock signal available on the REFO (RE3) pin. The RODIV<3:0> bits enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<5:4>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on RE3 when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for an EC or HS mode; otherwise, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode.

Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

REGISTER 3-4: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON	—	ROSSLP	ROSEL ⁽¹⁾	RODIV3	RODIV2	RODIV1	RODIV0
bit 7							bit 0

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

- bit 7 **ROON:** Reference Oscillator Output Enable bit
1 = Reference oscillator output is available on REFO pin
0 = Reference oscillator output is disabled
- bit 6 **Unimplemented:** Read as '0'
- bit 5 **ROSSLP:** Reference Oscillator Output Stop in Sleep bit
1 = Reference oscillator continues to run in Sleep
0 = Reference oscillator is disabled in Sleep
- bit 4 **ROSEL:** Reference Oscillator Source Select bit⁽¹⁾
1 = Primary oscillator (EC or HS) is used as the base clock
0 = System clock is used as the base clock; base clock reflects any clock switching of the device
- bit 3-0 **RODIV<3:0>:** Reference Oscillator Divisor Select bits
1111 = Base clock value divided by 32,768
1110 = Base clock value divided by 16,384
1101 = Base clock value divided by 8,192
1100 = Base clock value divided by 4,096
1011 = Base clock value divided by 2,048
1010 = Base clock value divided by 1,024
1001 = Base clock value divided by 512
1000 = Base clock value divided by 256
0111 = Base clock value divided by 128
0110 = Base clock value divided by 64
0101 = Base clock value divided by 32
0100 = Base clock value divided by 16
0011 = Base clock value divided by 8
0010 = Base clock value divided by 4
0001 = Base clock value divided by 2
0000 = Base clock value

Note 1: For ROSEL (REFOCON<4>), the primary oscillator is only available when configured as a default via the FOSC settings (regardless of whether the device is in Sleep mode).

12.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software-selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

The T0CON register (Register 12-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable.

Figure 12-1 provides a simplified block diagram of the Timer0 module in 8-bit mode. Figure 12-2 provides a simplified block diagram of the Timer0 module in 16-bit mode.

REGISTER 12-1: T0CON: TIMER0 CONTROL 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
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
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 **TMR0ON:** Timer0 On/Off Control bit

1 = Enables Timer0

0 = Stops Timer0

bit 6 **T08BIT:** Timer0 8-Bit/16-Bit Control bit

1 = Timer0 is configured as an 8-bit timer/counter

0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS:** Timer0 Clock Source Select bit

1 = Transition on T0CKI pin input edge

0 = Internal clock (Fosc/4)

bit 4 **T0SE:** Timer0 Source Edge Select bit

1 = Increment on high-to-low transition on the T0CKI pin

0 = Increment on low-to-high transition on the T0CKI pin

bit 3 **PSA:** Timer0 Prescaler Assignment bit

1 = Timer0 prescaler is not assigned; Timer0 clock input bypasses the prescaler

0 = Timer0 prescaler is assigned; Timer0 clock input comes from the prescaler output

bit 2-0 **T0PS<2:0>:** Timer0 Prescaler Select bits

111 = 1:256 Prescale value

110 = 1:128 Prescale value

101 = 1:64 Prescale value

100 = 1:32 Prescale value

011 = 1:16 Prescale value

010 = 1:8 Prescale value

001 = 1:4 Prescale value

000 = 1:2 Prescale value

13.4 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes. When the RD16 control bit (T1CON<1>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L loads the contents of the high byte of Timer1 into the Timer1 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer1 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 Timer1 must also take place through the TMR1H Buffer register. The Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits at once to both the high and low bytes of Timer1.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler; the prescaler is only cleared on writes to TMR1L.

13.5 SOSC Oscillator

An on-chip crystal oscillator circuit is incorporated between pins, SOSCI (input) and SOSCO (amplifier output). It is enabled by setting one of five bits: any of the four SOSCEN bits in the TxCON registers (TxCON<3>) or the SOSCGO bit in the OSCCON2 register (OSCCON2<3>). The oscillator is a low-power circuit, rated for 32 kHz crystals. It will continue to run during all power-managed modes. The circuit for a typical LP oscillator is depicted in Figure 13-2. Table 13-2 provides the capacitor selection for the SOSC oscillator.

The user must provide a software time delay to ensure proper start-up of the SOSC oscillator.

FIGURE 13-2: EXTERNAL COMPONENTS FOR THE SOSC OSCILLATOR

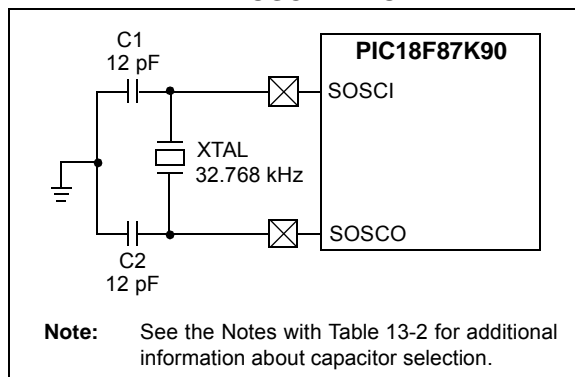


TABLE 13-2: CAPACITOR SELECTION FOR THE TIMER OSCILLATOR^(2,3,4,5)

Oscillator Type	Freq.	C1	C2
LP	32 kHz	12 pF ⁽¹⁾	12 pF ⁽¹⁾

Note 1: Microchip suggests these values as a starting point in validating the oscillator circuit.

2: Higher capacitance increases the stability of the oscillator, but also increases the start-up time.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

4: Capacitor values are for design guidance only. Values listed would be typical of a CL = 10 pF rated crystal when SOSCSEL<1:0> = 11.

5: Incorrect capacitance value may result in a frequency not meeting the crystal manufacturer's tolerance specification.

The SOSC crystal oscillator drive level is determined based on the SOSCSEL<1:0> (CONFIG1L<4:3>) Configuration bits. The High Drive Level mode, SOSCSEL<1:0> = 11, is intended to drive a wide variety of 32.768 kHz crystals with a variety of load capacitance (CL) ratings.

The Low Drive Level mode is highly optimized for extremely low-power consumption. It is not intended to drive all types of 32.768 kHz crystals. In the Low Drive Level mode, the crystal oscillator circuit may not work correctly if excessively large discrete capacitors are placed on the SOSCO and SOSCI pins. This mode is designed to work only with discrete capacitances of approximately 3 pF-10 pF on each pin.

Crystal manufacturers usually specify a CL (Capacitance Load) rating for their crystals. This value is related to, but not necessarily the same as, the values that should be used for C1 and C2 in Figure 13-2.

For more details on selecting the optimum C1 and C2 for a given crystal, see the crystal manufacturer's applications information. The optimum value depends, in part, on the amount of parasitic capacitance in the circuit, which is often unknown. For that reason, it is highly recommended that thorough testing and validation of the oscillator be performed after values have been selected.

PIC18F87K90 FAMILY

16.2 Timer4/6/8/10/12 Interrupt

The Timer4/6/8/10/12 modules have 8-bit Period registers, PRx, that are both readable and writable. Timer4/6/8/10/12 increment from 00h until they match PR4/6/8/10/12 and then reset to 00h on the next increment cycle. The PRx registers are initialized to FFh upon Reset.

16.3 Output of TMRx

The outputs of TMRx (before the postscaler) are used only as a PWM time base for the ECCP modules. They are not used as baud rate clocks for the MSSP modules as is the Timer2 output.

FIGURE 16-1: TIMER4/6/8/10/12 BLOCK DIAGRAM

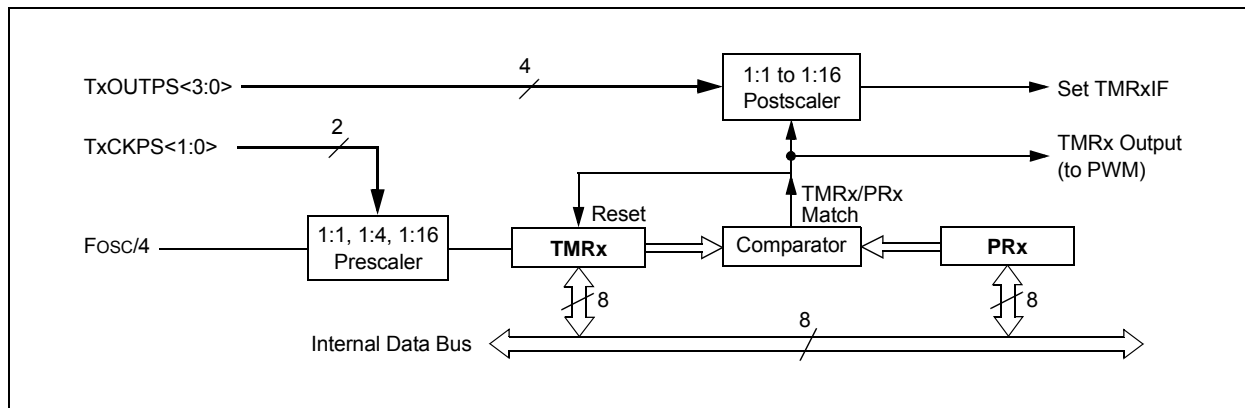


TABLE 16-3: REGISTERS ASSOCIATED WITH TIMER4/6/8/10/12 AS A TIMER/COUNTER

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	75
IPR5	TMR7GIP ⁽¹⁾	TMR12IP ⁽¹⁾	TMR10IP ⁽¹⁾	TMR8IP	TMR7IP ⁽¹⁾	TMR6IP	TMR5IP	TMR4IP	76
PIR5	TMR7GIF ⁽¹⁾	TMR12IF ⁽¹⁾	TMR10IF ⁽¹⁾	TMR8IF	TMR7IF ⁽¹⁾	TMR6IF	TMR5IF	TMR4IF	77
PIE5	TMR7GIE ⁽¹⁾	TMR12IE ⁽¹⁾	TMR10IE ⁽¹⁾	TMR8IE	TMR7IE ⁽¹⁾	TMR6IE	TMR5IE	TMR4IE	77
TMR4	Timer4 Register								82
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	82
PR4	Timer4 Period Register								82
TMR6	Timer6 Register								81
T6CON	—	T6OUTPS3	T6OUTPS2	T6OUTPS1	T6OUTPS0	TMR6ON	T6CKPS1	T6CKPS0	81
PR6	Timer6 Period Register								81
TMR8	Timer8 Register								81
T8CON	—	T8OUTPS3	T8OUTPS2	T8OUTPS1	T8OUTPS0	TMR8ON	T8CKPS1	T8CKPS0	81
PR8	Timer8 Period Register								81
TMR10	Timer10 Register								81
T10CON	—	T10OUTPS3	T10OUTPS2	T10OUTPS1	T10OUTPS0	TMR10ON	T10CKPS1	T10CKPS0	81
PR10	Timer10 Period Register								81
TMR12	Timer12 Register								81
T12CON	—	T12OUTPS3	T12OUTPS2	T12OUTPS1	T12OUTPS0	TMR12ON	T12CKPS1	T12CKPS0	81
PR12	Timer12 Period Register								81
CCPTMRS0	C3TSEL1	C3TSEL0	C2TSEL2	C2TSEL1	C2TSEL0	C1TSEL2	C1TSEL1	C1TSEL0	81
CCPTMRS1	C7TSEL1	C7TSEL0	—	C6TSEL0	—	C5TSEL0	C4TSEL1	C4TSEL0	81
CCPTMRS2	—	—	—	C10TSEL0 ⁽¹⁾	—	C9TSEL0 ⁽¹⁾	C8TSEL1	C8TSEL0	81

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer4/6/8/10/12 module.

Note 1: Unimplemented in devices with a program memory of 32 Kbytes (PIC18FX5K22).

PIC18F87K90 FAMILY

NOTES:

PIC18F87K90 FAMILY

17.1 RTCC MODULE REGISTERS

The RTCC module registers are divided into the following categories:

RTCC Control Registers

- RTCCFG
- RTCCAL
- PADCFG1
- ALRMCFG
- ALRMRPT

RTCC Value Registers

- RTCVALH
- RTCVALL

Both registers access the following registers:

- YEAR
- MONTH
- DAY
- WEEKDAY
- HOUR
- MINUTE
- SECOND

Alarm Value Registers

- ALRMVALH
- ALRMVALL

Both registers access the following registers:

- ALRMMNTH
- ALRMDAY
- ALRMWD
- ALRMHR
- ALRMMIN
- ALRMSEC

Note: The RTCVALH and RTCVALL registers can be accessed through RTCRPT<1:0> (RTCCFG<1:0>). ALRMVALH and ALRMVALL can be accessed through ALRMPTR<1:0> (ALRMCFG<1:0>).

17.1.4 RTCEN BIT WRITE

RTCWREN (RTCCFG<5>) must be set before a write to RTCEN can take place. Any write to the RTCEN bit, while RTCWREN = 0, will be ignored.

Like the RTCEN bit, the RTCVALH and RTCVALL registers can only be written to when RTCWREN = 1. A write to these registers, while RTCWREN = 0, will be ignored.

17.2 Operation

17.2.1 REGISTER INTERFACE

The register interface for the RTCC and alarm values is implemented using the Binary Coded Decimal (BCD) format. This simplifies the firmware when using the module as each of the digits is contained within its own 4-bit value (see Figure 17-2 and Figure 17-3).

FIGURE 17-2: TIMER DIGIT FORMAT

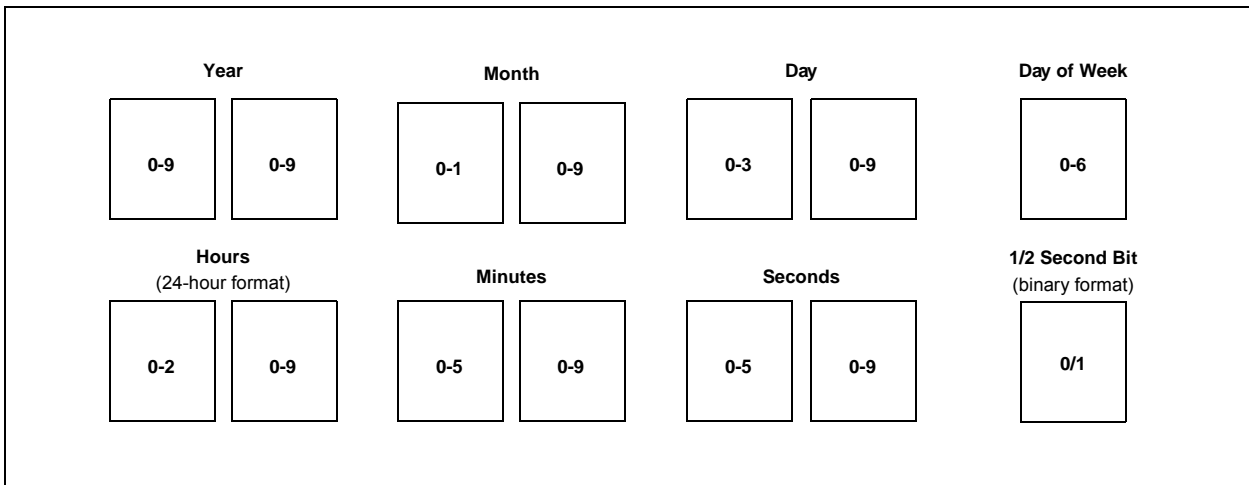
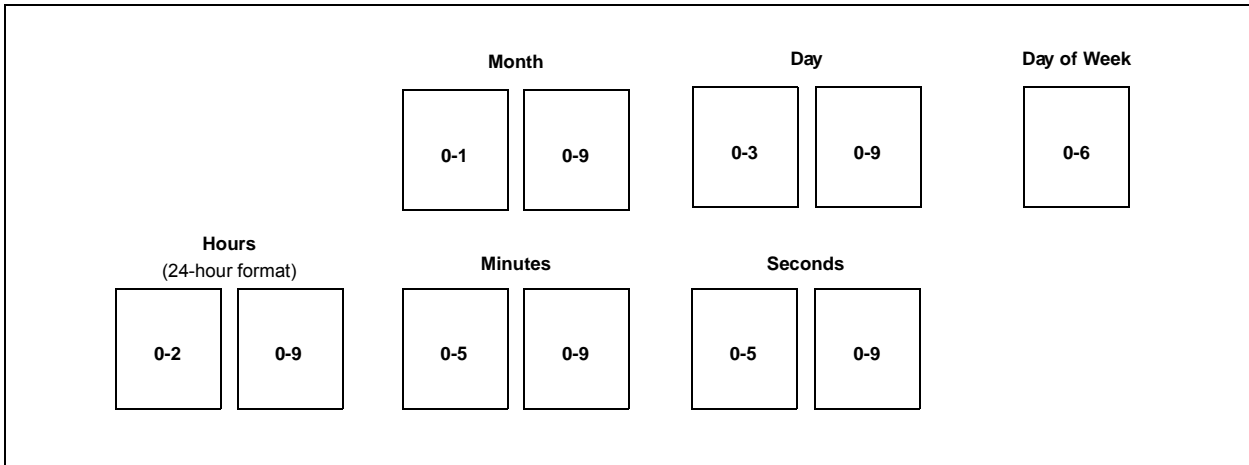


FIGURE 17-3: ALARM DIGIT FORMAT



17.2.4 LEAP YEAR

Since the year range on the RTCC module is 2000 to 2099, the leap year calculation is determined by any year divisible by four in the above range. Only February is affected in a leap year.

February will have 29 days in a leap year and 28 days in any other year.

17.2.5 GENERAL FUNCTIONALITY

All Timer registers containing a time value of seconds or greater are writable. The user configures the time by writing the required year, month, day, hour, minutes and seconds to the Timer registers, via Register Pointers. (See **Section 17.2.8 “Register Mapping”**.)

The timer uses the newly written values and proceeds with the count from the required starting point.

The RTCC is enabled by setting the RTCEN bit (RTCCFG<7>). If enabled, while adjusting these registers, the timer still continues to increment. However, any time the MINSEC register is written to, both of the timer prescalers are reset to '0'. This allows fraction of a second synchronization.

The Timer registers are updated in the same cycle as the write instruction's execution by the CPU. The user must ensure that when RTCEN = 1, the updated registers will not be incremented at the same time. This can be accomplished in several ways:

- By checking the RTCSYNC bit (RTCCFG<4>)
- By checking the preceding digits from which a carry can occur
- By updating the registers immediately following the seconds pulse (or an alarm interrupt)

The user has visibility to the half-second field of the counter. This value is read-only and can be reset only by writing to the lower half of the SECONDS register.

17.2.6 SAFETY WINDOW FOR REGISTER READS AND WRITES

The RTCSYNC bit indicates a time window during which the RTCC Clock Domain registers can be safely read and written without concern about a rollover. When RTCSYNC = 0, the registers can be safely accessed by the CPU.

Whether RTCSYNC = 1 or 0, the user should employ a firmware solution to ensure that the data read did not fall on a rollover boundary, resulting in an invalid or partial read. This firmware solution would consist of reading each register twice and then comparing the two values. If the two values matched, then a rollover did not occur.

17.2.7 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RTCCFG<5>) must be set.

To avoid accidental writes to the RTCC Timer register, it is recommended that the RTCWREN bit (RTCCFG<5>) be kept clear when not writing to the register. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN. For that reason, it is recommended that users follow the code example in Example 17-1.

EXAMPLE 17-1: SETTING THE RTCWREN BIT

```
movlw    0x55
movwf    EECON2
movlw    0xAA
movwf    EECON2
bsf      RTCCFG, RTCWREN
```

17.2.8 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Timer registers are accessed through corresponding Register Pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTRx bits (RTCCFG<1:0>) to select the required Timer register pair.

By reading or writing to the RTCVALH register, the RTCC Pointer value (RTCPTR<1:0>) decrements by '1' until it reaches '00'. When '00' is reached, the MINUTES and SECONDS value is accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 17-3: RTCVALH AND RTCVALL REGISTER MAPPING

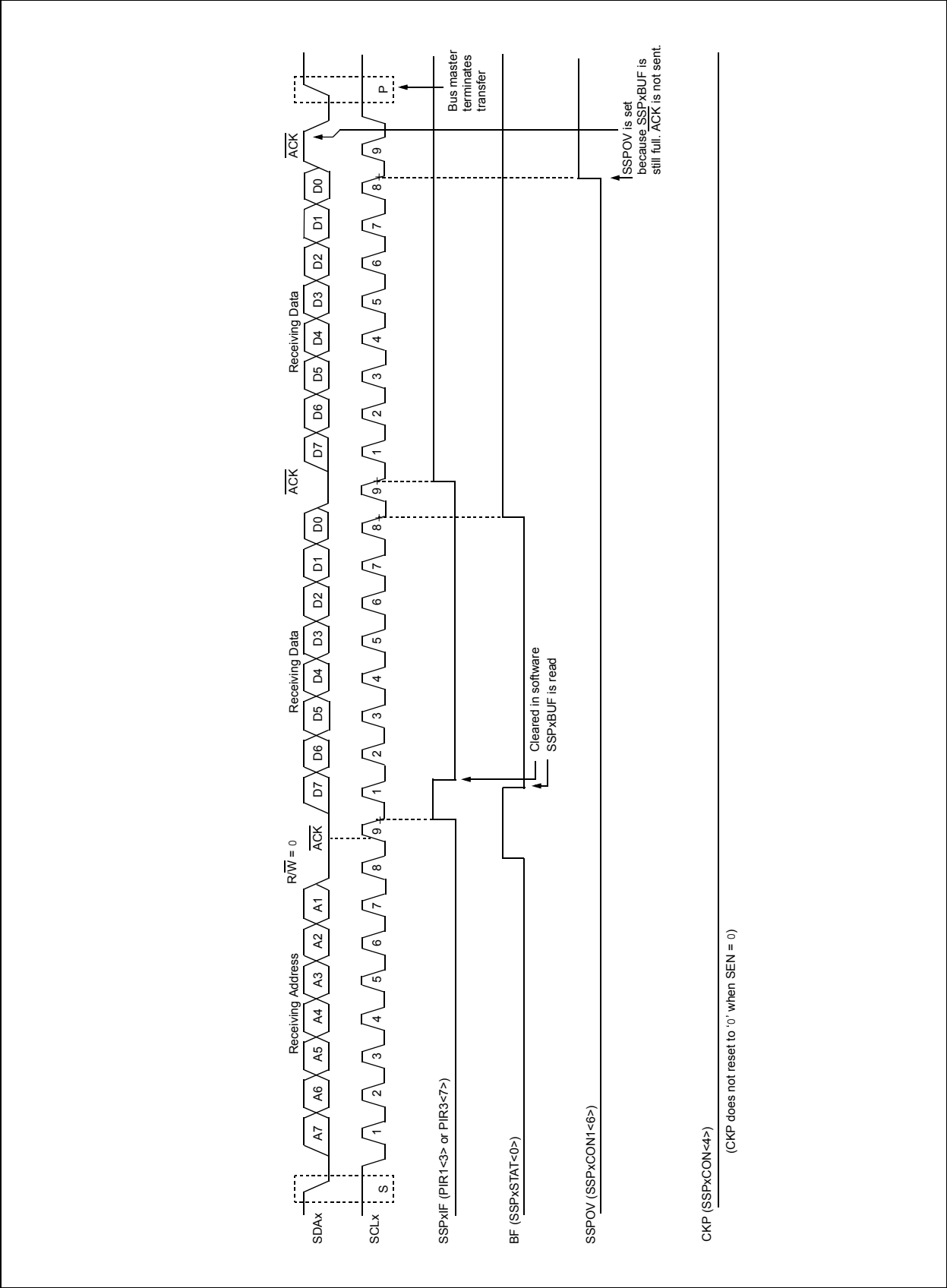
RTCPTR<1:0>	RTCC Value Register Window	
	RTCVALH	RTCVALL
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register windows (ALRMVALH and ALRMVALL) use the ALRMPTR bits (ALRMCFG<1:0>) to select the desired alarm register pair.

By reading or writing to the ALRMVALH register, the Alarm Pointer value, ALRMPTR<1:0>, decrements by one until it reaches '00'. When it reaches '00', the ALRMMIN and ALRMSEC value is accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

PIC18F87K90 FAMILY

FIGURE 21-8: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)



PIC18F87K90 FAMILY

The ANCONx registers are used to configure the operation of the I/O pin associated with each analog channel. Clearing a ANSELx bit configures the corresponding pin (ANx) to operate as a digital only I/O. Setting a bit configures the pin to operate as an analog

input for either the A/D Converter or the comparator module, with all digital peripherals disabled and digital inputs read as '0'.

As a rule, I/O pins that are multiplexed with analog inputs default to analog operation on any device Reset.

REGISTER 23-8: ANCON0: A/D PORT CONFIGURATION REGISTER 0

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
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<7:0>: Analog Port Configuration bits (AN7 and AN0)

0 = Pin is configured as a digital port

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

REGISTER 23-9: ANCON1: A/D PORT CONFIGURATION REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSEL15 ⁽¹⁾	ANSEL14 ⁽¹⁾	ANSEL13 ⁽¹⁾	ANSEL12 ⁽¹⁾	ANSEL11	ANSEL10	ANSEL9	ANSEL8
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<15:8>: Analog Port Configuration bits (AN15 through AN8)

0 = Pin is configured as a digital port

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

Note 1: AN12 through AN15, and AN20 to AN23, 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.

PIC18F87K90 FAMILY

NOTES:

PIC18F87K90 FAMILY

REGISTER 28-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	BORPWR1 ⁽¹⁾	BORPWR0 ⁽¹⁾	BORV1 ⁽¹⁾	BORV0 ⁽¹⁾	BOREN1 ⁽²⁾	BOREN0 ⁽²⁾	PWRTEN ⁽²⁾
bit 7							bit 0

Legend:	P = Programmable bit
R = Readable bit	W = Writable bit
-n = Value at POR	'1' = Bit is set
	'0' = Bit is cleared
	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-5 **BORPWR<1:0>:** BORMV Power Level bits⁽¹⁾

- 11 = ZPBORVMV instead of BORMV is selected
- 10 = BORMV is set to high-power level
- 01 = BORMV is set to medium-power level
- 00 = BORMV is set to low-power level

bit 4-3 **BORV<1:0>:** Brown-out Reset Voltage bits⁽¹⁾

- 11 = VBORMV is set to 1.8V
- 10 = VBORMV is set to 2.0V
- 01 = VBORMV is set to 2.7V
- 00 = VBORMV is set to 3.0V

bit 2-1 **BOREN<1:0>:** Brown-out Reset Enable bits⁽²⁾

- 11 = Brown-out Reset is enabled in hardware only (SBOREN is disabled)
- 10 = Brown-out Reset is enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)
- 01 = Brown-out Reset is enabled and controlled by software (SBOREN is enabled)
- 00 = Brown-out Reset is disabled in hardware and software

bit 0 **PWRTEN:** Power-up Timer Enable bit⁽²⁾

- 1 = PWRT is disabled
- 0 = PWRT is enabled

Note 1: For the specifications, see **Section 31.1 “DC Characteristics: Supply Voltage PIC18F87K90 Family (Industrial/Extended)”**.

2: The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

PIC18F87K90 FAMILY

REGISTER 28-10: CONFIG6L: CONFIGURATION REGISTER 6 LOW (BYTE ADDRESS 30000Ah)⁽²⁾

R/C-1	R/C-1	R/C-1	R/C-1	R/C-1	R/C-1	R/C-1	R/C-1
WRT7 ⁽¹⁾	WRT6 ⁽¹⁾	WRT5 ⁽¹⁾	WRT4 ⁽¹⁾	WRT3	WRT2	WRT1	WRT0
bit 7							bit 0

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

bit 7	WRT7: Write Protection bit ⁽¹⁾ 1 = Block 7 is not write-protected 0 = Block 7 is write-protected
bit 6	WRT6: Write Protection bit ⁽¹⁾ 1 = Block 6 is not write-protected 0 = Block 6 is write-protected
bit 5	WRT5: Write Protection bit ⁽¹⁾ 1 = Block 5 is not write-protected 0 = Block 5 is write-protected
bit 4	WRT4: Write Protection bit ⁽¹⁾ 1 = Block 4 is not write-protected 0 = Block 4 is write-protected
bit 3	WRT3: Write Protection bit 1 = Block 3 is not write-protected 0 = Block 3 is write-protected
bit 2	WRT2: Write Protection bit 1 = Block 2 is not write-protected 0 = Block 2 is write-protected
bit 1	WRT1: Write Protection bit 1 = Block 1 is not write-protected 0 = Block 1 is write-protected
bit 0	WRT0: Write Protection bit 1 = Block 0 is not write-protected 0 = Block 0 is write-protected

Note 1: This bit is only available on PIC18F67K90 and PIC18F87K90.

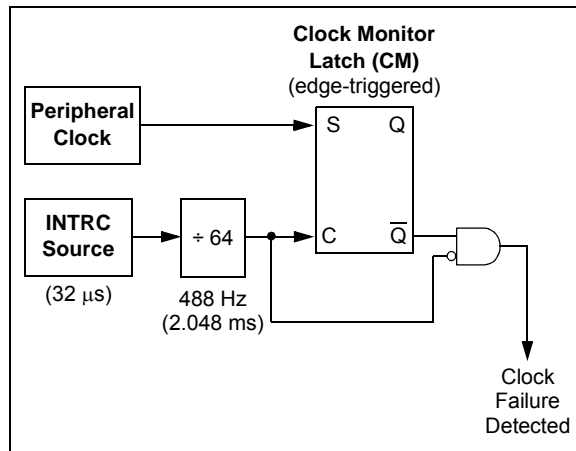
2: For the memory size of the blocks, refer to Figure 28-6.

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.

PIC18F87K90 FAMILY

TABLE 29-2: PIC18F87K90 FAMILY INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
			MSb		LSb			
LITERAL OPERATIONS								
ADDLW k	Add Literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	AND Literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW k	Inclusive OR Literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR f, k	Move literal (12-bit) 2nd word to FSR(f) 1st word	2	1110	1110	00ff	kkkk	None	
MOVLB k	Move Literal to BSR<3:0>	1	0000	0001	0000	kkkk	None	
MOVLW k	Move Literal to WREG	1	0000	1110	kkkk	kkkk	None	
MULLW k	Multiply Literal with WREG	1	0000	1101	kkkk	kkkk	None	
RETLW k	Return with Literal in WREG	2	0000	1100	kkkk	kkkk	None	
SUBLW k	Subtract WREG from Literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR Literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS								
TBLRD*	Table Read	2	0000	0000	0000	1000	None	
TBLRD*+	Table Read with Post-Increment		0000	0000	0000	1001	None	
TBLRD*-	Table Read with Post-Decrement	0000	0000	0000	1010	None		
TBLRD*+	Table Read with Pre-Increment	0000	0000	0000	1011	None		
TBLWT*	Table Write	2	0000	0000	0000	1100	None	
TBLWT*+	Table Write with Post-Increment		0000	0000	0000	1101	None	
TBLWT*-	Table Write with Post-Decrement		0000	0000	0000	1110	None	
TBLWT*+	Table Write with Pre-Increment		0000	0000	0000	1111	None	

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVWF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable, $d = 1$), the prescaler will be cleared if assigned.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

PIC18F87K90 FAMILY

ANDWF

AND W with f

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

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

0001	01da	ffff	ffff
------	------	------	------

Description: The contents of W are ANDed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

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 $f \leq 95$ (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 to destination

Example: ANDWF REG, 0, 0

Before Instruction

W = 17h
 REG = C2h

After Instruction

W = 02h
 REG = C2h

BC

Branch if Carry

Syntax: BC n

Operands: $-128 \leq n \leq 127$

Operation: if Carry bit is '1',
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0010	nnnn	nnnn
------	------	------	------

Description: If the Carry bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BC 5

Before Instruction

PC = address (HERE)

After Instruction

If Carry = 1;
 PC = address (HERE + 12)
 If Carry = 0;
 PC = address (HERE + 2)