Welcome to E-XFL.COM

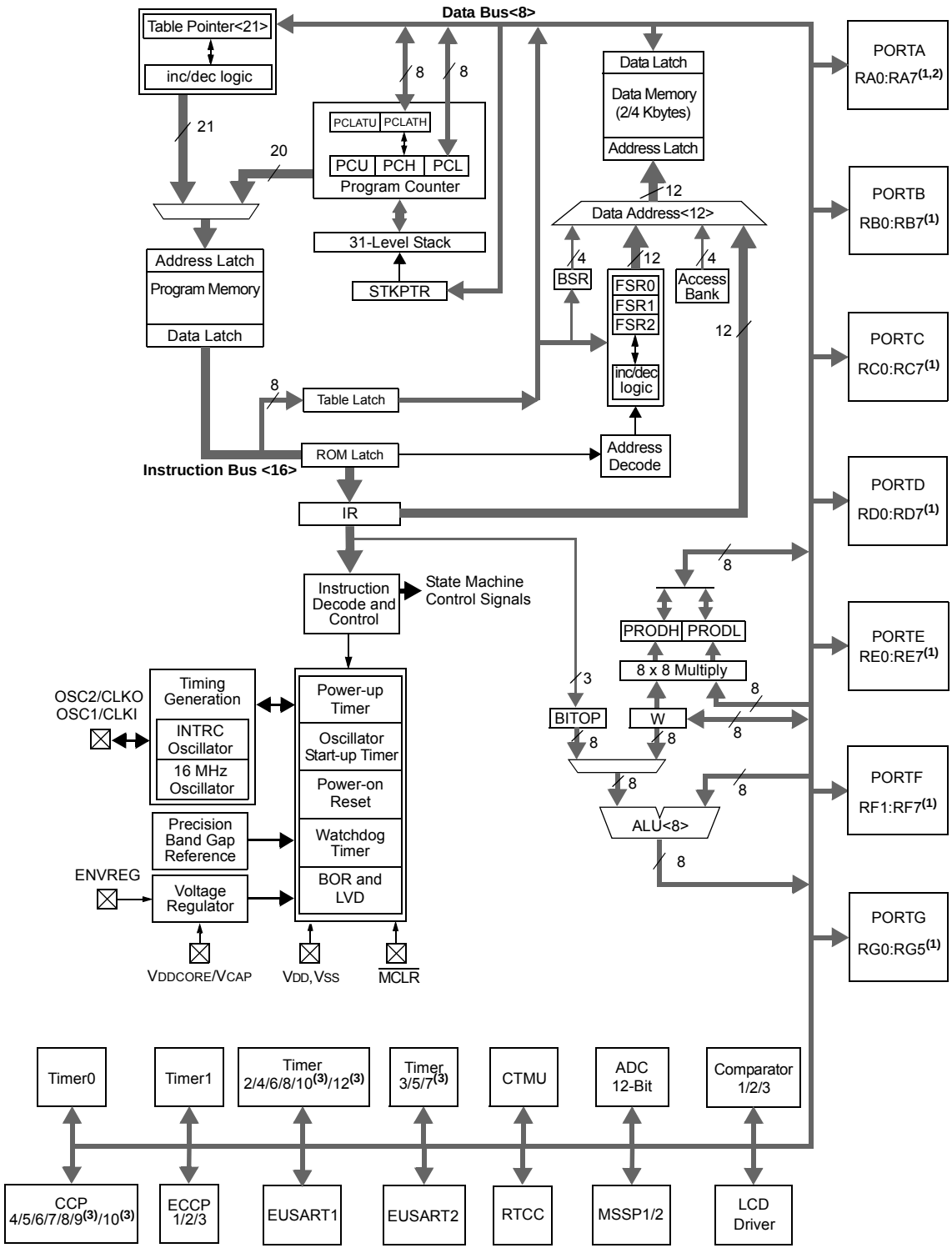
What is "Embedded - Microcontrollers"?

"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, LCD, POR, PWM, WDT
Number of I/O	69
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 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f86k90t-i-pt

FIGURE 1-1: PIC18F6XK90 (64-PIN) BLOCK DIAGRAM

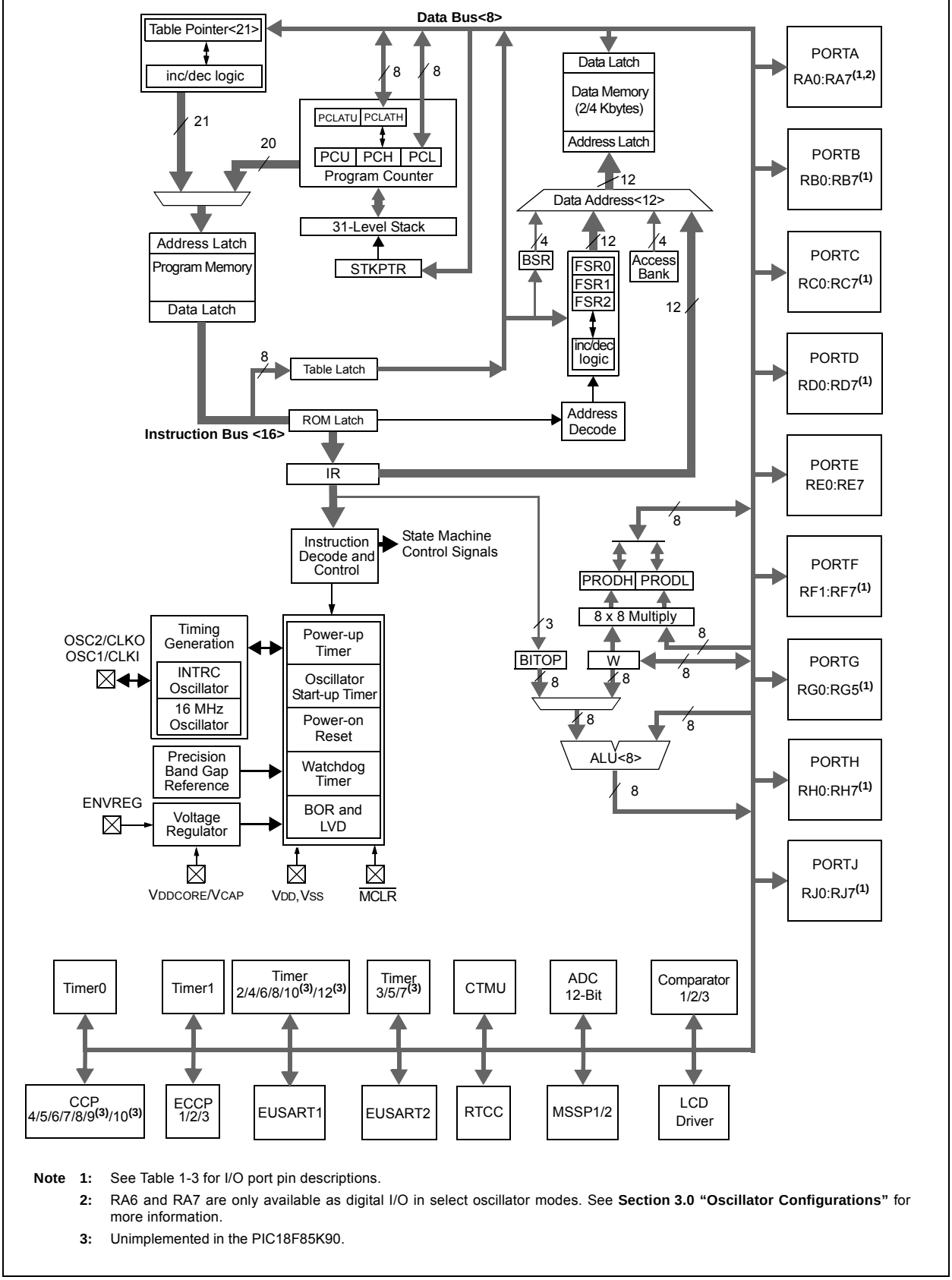


Note 1: See Table 1-3 for I/O port pin descriptions.

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

3: Unimplemented in the PIC18F65K90.

FIGURE 1-2: PIC18F8XK90 (80-PIN) BLOCK DIAGRAM

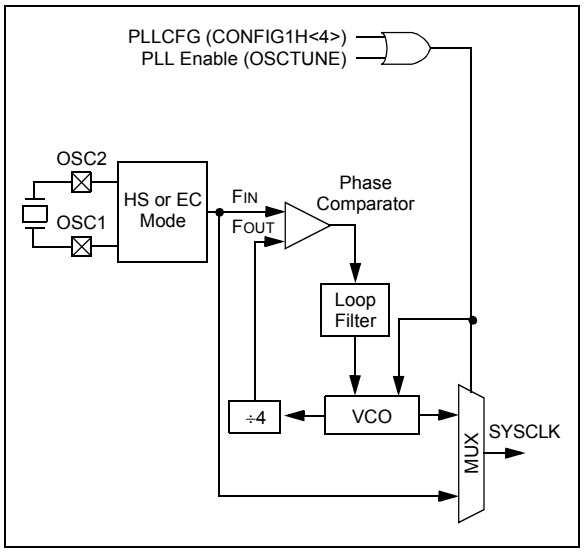


Note 1: See Table 1-3 for I/O port pin descriptions.

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

3: Unimplemented in the PIC18F85K90.

FIGURE 3-7: PLL BLOCK DIAGRAM



3.5.2.2 PLL and HF-INTOSC

The PLL is available to the internal oscillator block when the internal oscillator block is configured as the primary clock source. In this configuration, the PLL is enabled in software and generates a clock output of up to 64 MHz.

The operation of INTOSC with the PLL is described in **Section 3.6.2 "INTPLL Modes"**. Care should be taken that the PLL is enabled only if the HF-INTOSC postscaler is configured for 8 MHz or 16 MHz.

3.6 Internal Oscillator Block

The PIC18F87K90 family of devices includes an internal oscillator block which generates two different clock signals. Either clock can be used as the microcontroller's clock source, which may eliminate the need for an external oscillator circuit on the OSC1 and/or OSC2 pins.

The internal oscillator consists of three blocks, depending on the frequency of operation. They are HF-INTOSC, MF-INTOSC and LF-INTRC.

In HF-INTOSC mode, the internal oscillator can provide a frequency, ranging from 31 kHz to 16 MHz, with the postscaler deciding the selected frequency (IRCF<2:0>).

The INTSRC bit (OSCTUNE<7>) and MFIOSEL bit (OSCCON2<0>) also decide which INTOSC provides the lower frequency (500 kHz to 31 kHz). For the HF-INTOSC to provide these frequencies, INTSRC = 1 and MFI0SEL = 0.

In HF-INTOSC, the postscaler (IRCF<2:0>) provides the frequency range of 31 kHz to 16 MHz. If HF-INTOSC is used with the PLL, the input frequency to the PLL should be 8 MHz or 16 MHz (IRCF<2:0> = 111, 110).

For MF-INTOSC mode to provide a frequency range of 500 kHz to 31 kHz, INTSRC = 1 and MFIOSEL = 1. The postscaler (IRCF<2:0>), in this mode, provides the frequency range of 31 kHz to 500 kHz.

The LF-INTRC can provide only 31 kHz if INTSRC = 0.

The LF-INTRC provides 31 kHz and is enabled if it is selected as the device clock source. The mode is enabled automatically when any of the following is enabled:

- Power-up Timer
- Fail-Safe Clock Monitor
- Watchdog Timer
- Two-Speed Start-up

These features are discussed in greater detail in **Section 28.0 "Special Features of the CPU"**.

The clock source frequency (HF-INTOSC, MF-INTOSC or LF-INTRC direct) is selected by configuring the IRCF bits of the OSCCON register, as well the INTSRC and MFIOSEL bits. The default frequency on device Resets is 8 MHz.

3.6.1 INTIO MODES

Using the internal oscillator as the clock source eliminates the need for up to two external oscillator pins, which can then be used for digital I/O. Two distinct oscillator configurations, which are determined by the OSC Configuration bits, are available:

- In INTIO1 mode, the OSC2 pin (RA6) outputs FOSC/4, while OSC1 functions as RA7 (see Figure 3-8) for digital input and output.
- In INTIO2 mode, OSC1 functions as RA7 and OSC2 functions as RA6 (see Figure 3-9). Both are available as digital input and output ports.

FIGURE 3-8: INTIO1 OSCILLATOR MODE

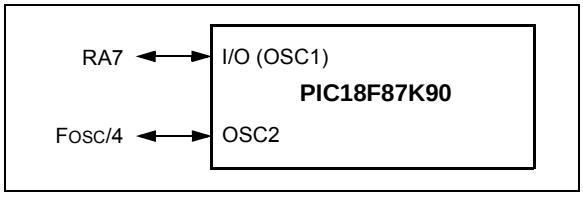


FIGURE 3-9: INTIO2 OSCILLATOR MODE

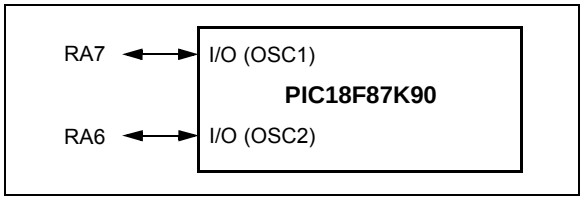


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

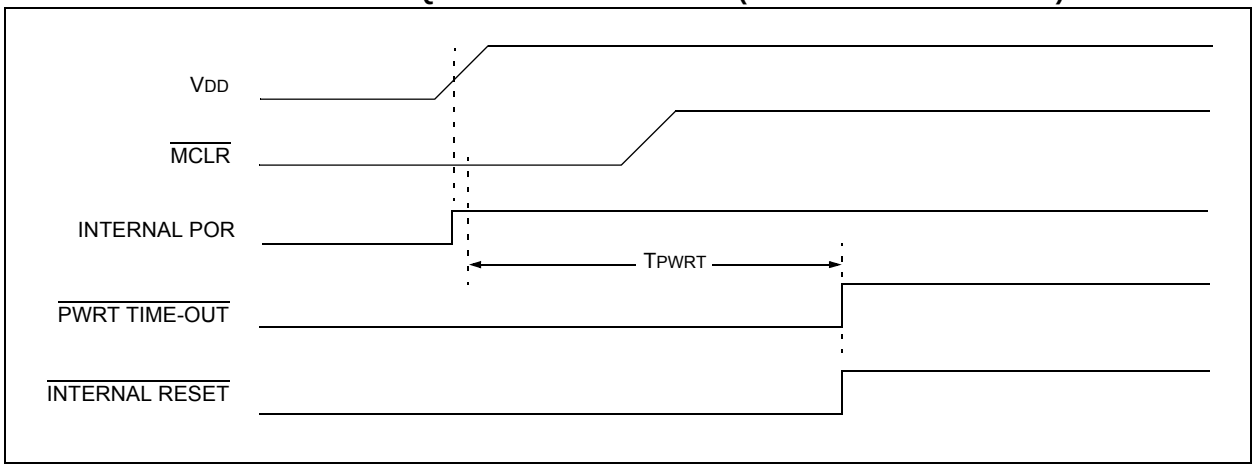


FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

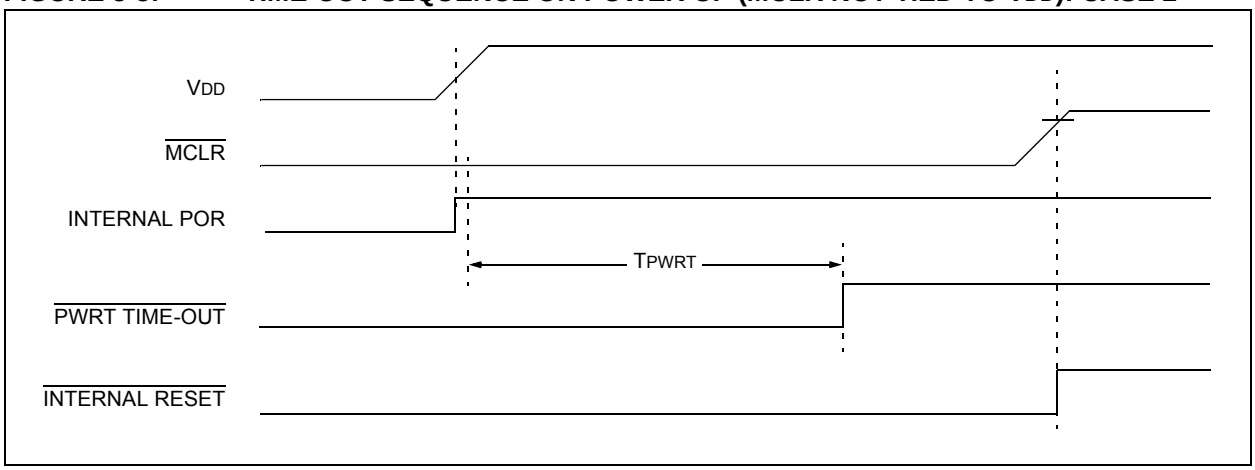


FIGURE 5-6: SLOW RISE TIME (MCLR TIED TO VDD, VDD RISE > TPWRT)

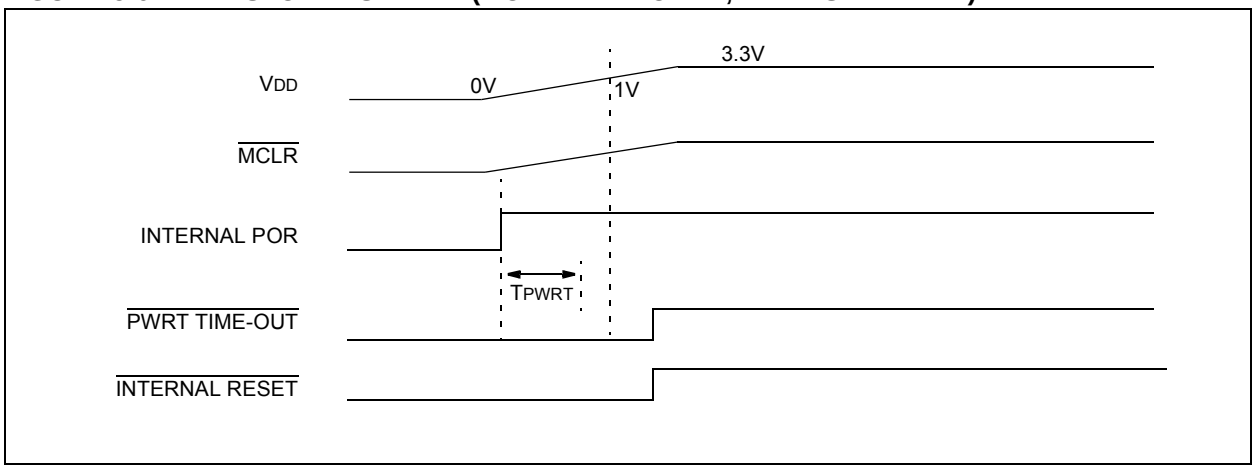


TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
CCPR10L	PIC18F6XK90	PIC18F8XK90	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP10CON	PIC18F6XK90	PIC18F8XK90	--00 0000	--00 0000	--uu uuuu
TMR7H	PIC18F6XK90	PIC18F8XK90	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR7L	PIC18F6XK90	PIC18F8XK90	xxxx xxxx	uuuu uuuu	uuuu uuuu
T7CON	PIC18F6XK90	PIC18F8XK90	0000 0000	uuuu uuuu	uuuu -uuu
T7GCON	PIC18F6XK90	PIC18F8XK90	0000 0x00	0000 0x00	uuuu uuuu
TMR6	PIC18F6XK90	PIC18F8XK90	0000 0000	0000 0000	uuuu uuuu
PR6	PIC18F6XK90	PIC18F8XK90	1111 1111	1111 1111	uuuu uuuu
T6CON	PIC18F6XK90	PIC18F8XK90	-000 0000	-000 0000	-uuu uuuu
TMR8	PIC18F6XK90	PIC18F8XK90	0000 0000	0000 0000	uuuu uuuu
PR8	PIC18F6XK90	PIC18F8XK90	1111 1111	1111 1111	uuuu uuuu
T8CON	PIC18F6XK90	PIC18F8XK90	-000 0000	-000 0000	-uuu uuuu
TMR10	PIC18F6XK90	PIC18F8XK90	0000 0000	0000 0000	uuuu uuuu
PR10	PIC18F6XK90	PIC18F8XK90	1111 1111	1111 1111	uuuu uuuu
T10CON	PIC18F6XK90	PIC18F8XK90	-000 0000	-000 0000	-uuu uuuu
TMR12	PIC18F6XK90	PIC18F8XK90	0000 0000	0000 0000	uuuu uuuu
PR12	PIC18F6XK90	PIC18F8XK90	1111 1111	1111 1111	uuuu uuuu
T12CON	PIC18F6XK90	PIC18F8XK90	-000 0000	-000 0000	-uuu uuuu
CM2CON	PIC18F6XK90	PIC18F8XK90	0001 1111	0001 1111	uuuu uuuu
CM3CON	PIC18F6XK90	PIC18F8XK90	0001 1111	0001 1111	uuuu uuuu
CCPTMRS0	PIC18F6XK90	PIC18F8XK90	0000 0000	uuuu uuuu	uuuu uuuu
CCPTMRS1	PIC18F6XK90	PIC18F8XK90	00-0 -000	uu-u -uuu	uu-u -uuu
CCPTMRS2	PIC18F6XK90	PIC18F8XK90	---0 -000	---u -uuu	---u -uuu
REFOCON	PIC18F6XK90	PIC18F8XK90	0-00 0000	u-uu uuuu	u-uu uuuu
ODCON1	PIC18F6XK90	PIC18F8XK90	000- ---0	uuu- ---u	uuu- ---u
ODCON2	PIC18F6XK90	PIC18F8XK90	0000 0000	uuuu uuuu	uuuu uuuu
ODCON3	PIC18F6XK90	PIC18F8XK90	00-- ---0	uu-- ---u	uu-- ---u
ANCON0	PIC18F6XK90	PIC18F8XK90	1111 1111	uuuu uuuu	uuuu uuuu
ANCON1	PIC18F6XK90	PIC18F8XK90	1111 1111	uuuu uuuu	uuuu uuuu
ANCON2	PIC18F6XK90	PIC18F8XK90	1111 1111	uuuu uuuu	uuuu uuuu
RCSTA2	PIC18F6XK90	PIC18F8XK90	0000 000x	0000 000x	uuuu uuuu
TXSTA2	PIC18F6XK90	PIC18F8XK90	0000 0010	0000 0010	uuuu uuuu
BAUDCON2	PIC18F6XK90	PIC18F8XK90	0100 0-00	0100 0-00	uuuu u-uu

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

Note 1: When the wake-up is due to an interrupt, and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

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

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 5-1 for the Reset value for a specific condition.

TABLE 6-1: PIC18F87K90 FAMILY SPECIAL FUNCTION REGISTER MAP(5) (CONTINUED)

Addr.	Name	Addr.	Name	Addr.	Name	Addr.	Name	Addr.	Name	Addr.	Name
F3Eh	TMR7L(4)	F31h	PR12(4)	F24h	ANCON1	F17h	PMD2	F0Ah	CCPR6L	EFDh	LCDREF
F3Dh	T7CON(4)	F30h	T12CON(4)	F23h	ANCON2	F16h	PMD3	F09h	CCP6CON	EFCh	LCDRL
F3Ch	T7GCON(4)	F2Fh	CM2CON	F22h	RCSTA2	F15h	TMR5H	F08h	CCPR7H	EFBh	LCDSE5(3)
F3Bh	TMR6	F2Eh	CM3CON	F21h	TXSTA2	F14h	TMR5L	F07h	CCPR7L	EFAh	LCDSE4
F3Ah	PR6	F2Dh	CCPTMRS0	F20h	BAUDCON2	F13h	T5CON	F06h	CCP7CON	EF9h	LCDSE3
F39H	T6CON	F2Ch	CCPTMRS1	F1Fh	SPBRGH2	F12h	T5GCON	F05h	TMR4	EF8h	LCDSE2
F38h	TMR8	F2Bh	CCPTMRS2	F1Eh	SPBRG2	F11h	CCPR4H	F04h	PR4	EF7h	LCDSE1
F37h	PR8	F2Ah	REFOCON	F1Dh	RCREG2	F10h	CCPR4L	F03h	T4CON	EF6h	LCDSE0
F36h	T8CON	F29H	ODCON1	F1Ch	TXREG2	F0Fh	CCP4CON	F02h	SSP2BUF	EF5h	LCDPS
F35h	TMR10(4)	F28h	ODCON2	F1Bh	PSTR2CON	F0Eh	CCPR5H	F01h	SSP2ADD	EF4h	LCDCON
F34h	PR10(4)	F27h	ODCON3	F1Ah	PSTR3CON	F0Dh	CCPR5L	F00h	SSP2STAT		
F33h	T10CON(4)	F26h	—	F19h	PMD0	F0Ch	CCP5CON	EFFh	SSP2CON1		

Note 1: This is not a physical register.
2: Unimplemented registers are read as '0'.
3: This register is not available in 64-pin devices (PIC18F6XK90).
4: This register is not available in devices with a program memory of 32 Kbytes (PIC18FX5K90).
5: Addresses, EF4h through F5Fh, are also used by SFRs, but are not part of the Access RAM. Users must always load the proper BSR value to access these registers.

REGISTER 8-1: EECON1: DATA EEPROM CONTROL REGISTER 1

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR(1)	WREN	WR	RD
bit 7							bit 0

Legend:	S = Settable 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 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
1 = Access Flash program memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
1 = Access Configuration registers
0 = Access Flash program or data EEPROM memory

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

bit 4 **FREE:** Flash Row Erase Enable bit
1 = Erase the program memory row addressed by the TBLPTR on the next WR command (cleared by completion of an erase operation)
0 = Perform write-only

bit 3 **WRERR:** Flash Program/Data EEPROM Error Flag bit(1)
1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation or an improper write attempt)
0 = The write operation completed

bit 2 **WREN:** Flash Program/Data EEPROM Write Enable bit
1 = Allows write cycles to Flash program/data EEPROM
0 = Inhibits write cycles to Flash program/data EEPROM

bit 1 **WR:** Write-Control bit
1 = Initiates a data EEPROM erase/write cycle, or a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once the write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. The RD bit cannot be set when EEPGD = 1 or CFGS = 1.)
0 = Does not initiate an EEPROM read

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

REGISTER 10-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	—	SSP2IF	BCL2IF	BCL1IF	HLVDIF	TMR3IF	TMR3GIF
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 **OSCFIF:** Oscillator Fail Interrupt Flag bit
1 = Device oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = Device clock operating

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

bit 5 **SSP2IF:** Master Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception has been completed (must be cleared in software)
0 = Waiting to transmit/receive

bit 4 **BCL2IF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

bit 3 **BCL1IF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

bit 2 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag bit
1 = A high/low-voltage condition occurred (must be cleared in software)
0 = The device voltage is above the regulator's low-voltage trip point

bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **TMR3GIF:** TMR3 Gate Interrupt Flag bit
1 = Timer gate interrupt occurred (must be cleared in software)
0 = No timer gate interrupt occurred

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

Timer3/5/7 can be configured for 16-bit reads and writes (see Figure 15.3). When the RD16 control bit (TxCON<1>) is set, the address for TMRxH is mapped to a buffer register for the high byte of Timer3/5/7. A read from TMRxL will load the contents of the high byte of Timer3/5/7 into the Timerx High Byte Buffer register. This provides users with the ability to accurately read all 16 bits of Timer3/5/7 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

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

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

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

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

The SOSC internal oscillator may be used as the clock source for Timer3/5/7. The SOSC oscillator is enabled by 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>). To use it as the Timer3/5/7 clock source, the TMRxCS bit must also be set. As previously noted, this also configures Timer3/5/7 to increment on every rising edge of the oscillator source.

The SOSC oscillator is described in **Section 13.0 "Timer1 Module"**.

15.5 Timer3/5/7 Gates

Timer3/5/7 can be configured to count freely or the count can be enabled and disabled using the Timer3/5/7 gate circuitry. This is also referred to as the Timer3/5/7 gate count enable.

The Timer3/5/7 gate can also be driven by multiple selectable sources.

15.5.1 TIMER3/5/7 GATE COUNT ENABLE

The Timerx Gate Enable mode is enabled by setting the TMRxGE bit (TxGCON<7>). The polarity of the Timerx Gate Enable mode is configured using the TxGPOL bit (TxGCON<6>).

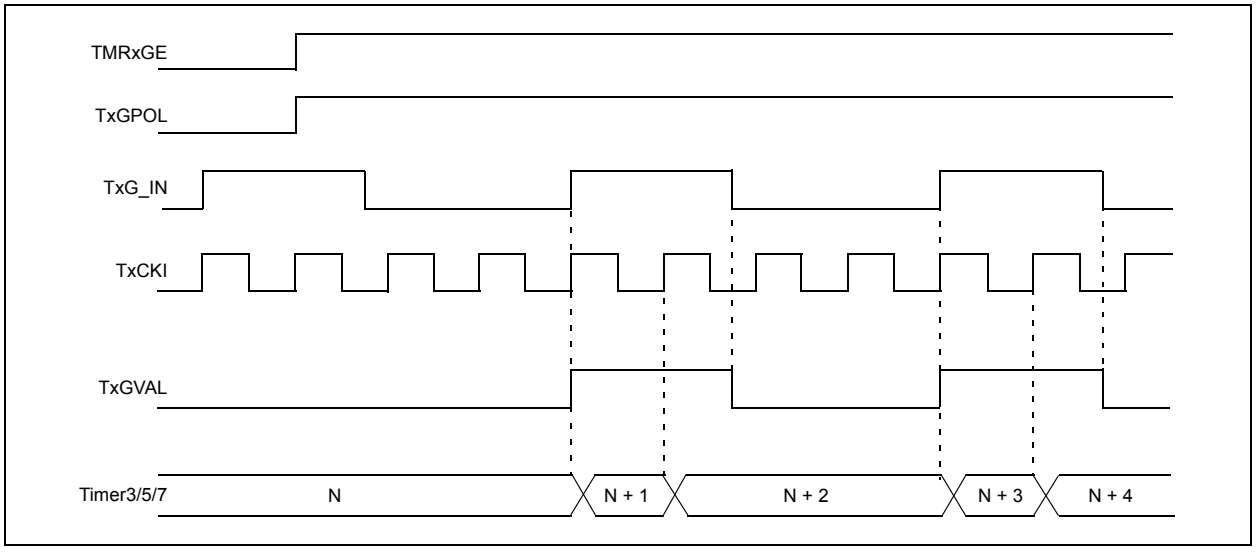
When Timerx Gate Enable mode is enabled, Timer3/5/7 will increment on the rising edge of the Timer3/5/7 clock source. When Timerx Gate Enable mode is disabled, no incrementing will occur and Timer3/5/7 will hold the current count. See Figure 15-2 for timing details.

TABLE 15-1: TIMER3/5/7 GATE ENABLE SELECTIONS

TxCLK(†)	TxGPOL (TxGCON<6>)	TxG Pin	Timerx Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
	1	1	Counts

† The clock on which TMR3/5/7 is running. For more information, see TxCLK in Figure 15-1.

FIGURE 15-2: TIMER3/5/7 GATE COUNT ENABLE MODE



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

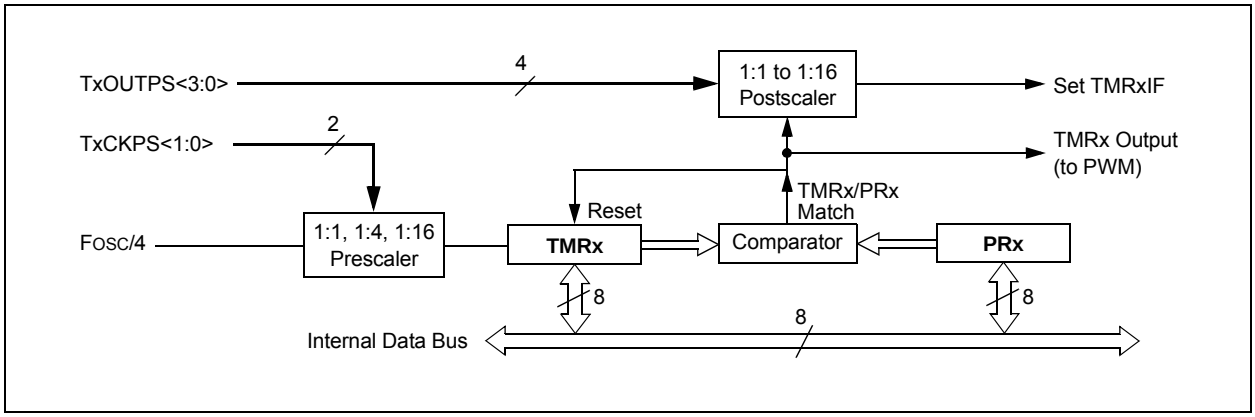


TABLE 16-3: REGISTERS ASSOCIATED WITH TIMER4/6/8/10/12 AS A TIMER/COUNTER

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).

REGISTER 17-7: YEAR: YEAR VALUE REGISTER(1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
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 7-4 **YRTEN<3:0>:** Binary Coded Decimal Value of Year's Tens Digit bits
Contains a value from 0 to 9.

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

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

REGISTER 17-8: MONTH: MONTH VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

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

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

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

17.1.3 ALRMVALH AND ALRMVALL REGISTER MAPPINGS

REGISTER 17-14: ALRMMNTH: ALARM MONTH VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

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

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

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

REGISTER 17-15: ALRMDAY: ALARM DAY VALUE REGISTER(1)

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

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

bit 7-6 **Unimplemented:** Read as '0'

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

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

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

19.4.8 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2/4/6/8 will not increment and the state of the module will not change. If the ECCPx pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from HF-INTOSC and the postscaler may not be immediately stable.

In PRI_IDLE mode, the primary clock will continue to clock the ECCPx module without change.

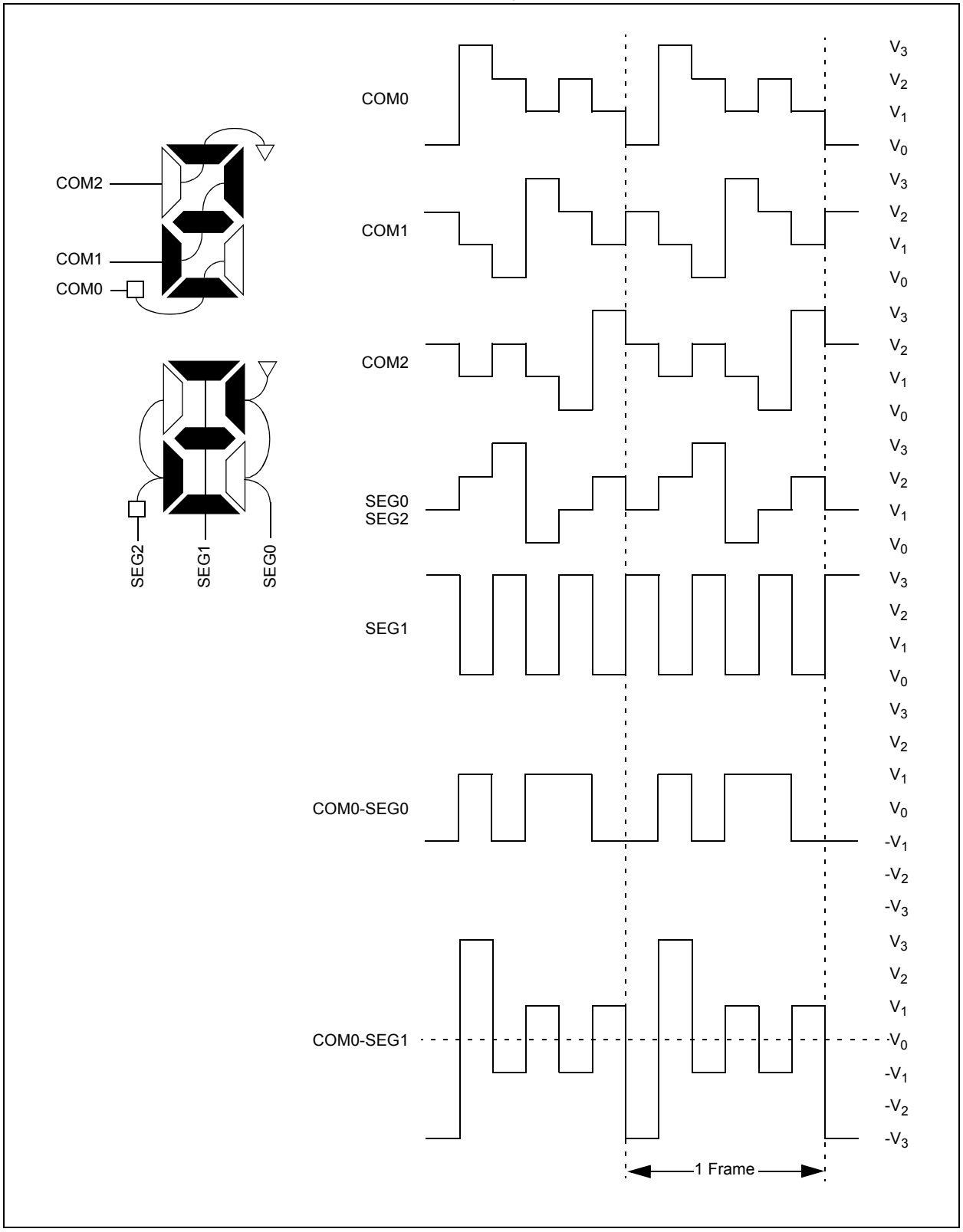
19.4.8.1 Operation with Fail-Safe Clock Monitor (FSCM)

If the Fail-Safe Clock Monitor (FSCM) is enabled, a clock failure will force the device into the power-managed RC_RUN mode and the OSCFIF bit of the PIR2/4/6/8 register will be set. The ECCPx will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

19.4.9 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the ECCP registers to their Reset states. This forces the ECCP module to reset to a state compatible with previous, non-Enhanced CCP modules used on other PIC18 and PIC16 devices.

FIGURE 20-14: TYPE-A WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE



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 MSSPx 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 MSSPx module shifts in the ACK bit from the slave device and writes its value into the SSPxCON2 register (SSPxCON2<6>).
6. The MSSPx module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
7. The user loads the SSPxBUF with 8 bits of data.
8. Data is shifted out the SDAx pin until all 8 bits are transmitted.
9. The MSSPx module shifts in the ACK bit from the slave device and writes its value into the SSPxCON2 register (SSPxCON2<6>).
10. The MSSPx 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. An interrupt is generated once the Stop condition is complete.

27.6 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. To do that:

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 $T = (C/I) \cdot V$, where:
   - I is calculated in the current calibration step (**Section 27.4.1 "Current Source Calibration"**)
   - C is calculated in the capacitance calibration step (**Section 27.4.2 "Capacitance Calibration"**)
   - 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 (25 pF).

To measure longer time intervals, an external capacitor may be connected to an A/D channel and that channel selected whenever making a time measurement.

FIGURE 27-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT

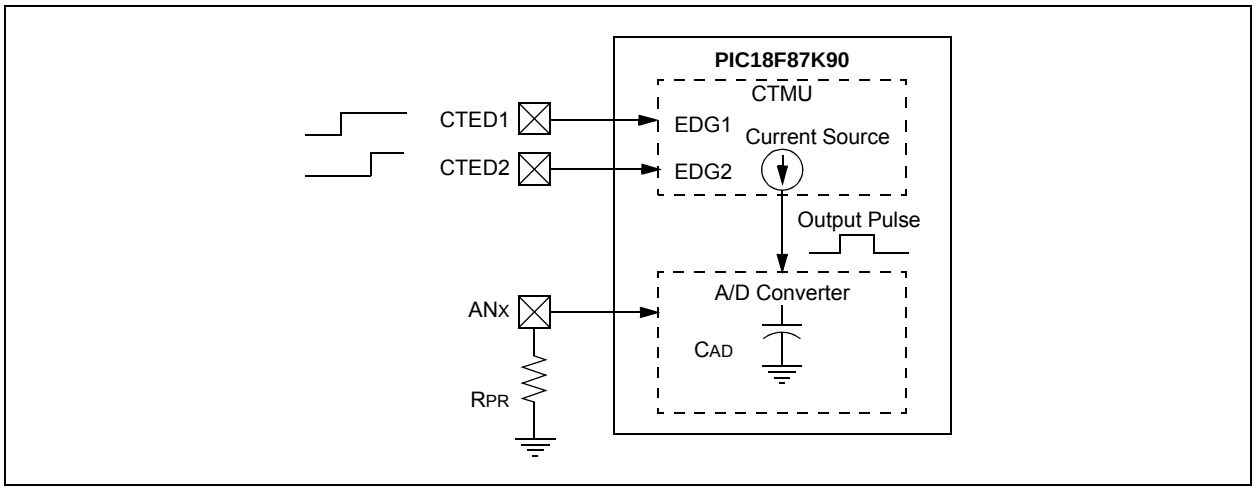


TABLE 28-1: CONFIGURATION BITS AND DEVICE IDs

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	—	XINST	—	SOSCSEL1	SOSCSEL0	INTOSCSEL	—	$\overline{\text{RETEN}}$	-1-1 1--1
300001h	CONFIG1H	IESO	FCMEN	—	PLLCFG	FOSC3	FOSC2	FOSC1	FOSC0	0000 1000
300002h	CONFIG2L	—	BORPWR1	BORWPR0	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	-111 1111
300003h	CONFIG2H	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN1	WDTEN0	-111 1111
300004h	CONFIG3L	—	—	—	—	—	—	—	RTCOSC	---- ---1
300005h	CONFIG3H	MCLRE	—	—	—	MSSPMSK	—	ECCPMX(2)	CCP2MX	1--- 1-11
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	—	—	BBSIZ0	—	—	—	STVREN	1--1 ---1
300008h	CONFIG5L	CP7(1)	CP6(1)	CP5(1)	CP4(1)	CP3	CP2	CP1	CP0	1111 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	WRT7(1)	WRT6(1)	WRT5(1)	WRT4(1)	WRT3	WRT2	WRT1	WRT0	1111 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	EBTR7(1)	EBTR6(1)	EBTR5(1)	EBTR4(1)	EBTR3	EBTR2	EBTR1	EBTR0	1111 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1(3)	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2(3)	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	xxxx xxxx

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition.
Shaded cells are unimplemented, read as '0'.

Note 1: Implemented in the PIC18F67K90 and PIC18F87K90 devices.

2: Implemented in the 80-pin devices (PIC18F8XK90).

3: See Register 28-14 for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.

RETURN **Return from Subroutine**

Syntax: RETURN {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC;
if s = 1,
(WS) → W,
(STATUSS) → STATUS,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	0000	0001	001s

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the Program Counter. If 's'= 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	POP PC from stack
No operation	No operation	No operation	No operation

Example: RETURN

After Instruction:
PC = TOS

RLCF **Rotate Left f through Carry**

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

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

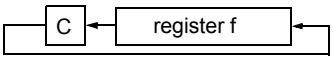
Encoding:

0011	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed 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 ≤ 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: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

29.2.2 EXTENDED INSTRUCTION SET

ADDFSR **Add Literal to FSR**

Syntax: ADDFSR f, k

Operands: $0 \le k \le 63$
$f \in [0, 1, 2]$

Operation: FSR(f) + k → FSR(f)

Status Affected: None

Encoding:	1110	1000	ffkk	kkkk

Description: The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR

Example: ADDFSR 2, 23h

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

ADDULNK **Add Literal to FSR2 and Return**

Syntax: ADDULNK k

Operands: $0 \le k \le 63$

Operation: FSR2 + k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:	1110	1000	11kk	kkkk

Description: The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a NOP is performed during the second cycle.

This may be thought of as a special case of the ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: ADDULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 0422h
PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

INDEX

A

B