



Welcome to [E-XFL.COM](https://www.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.

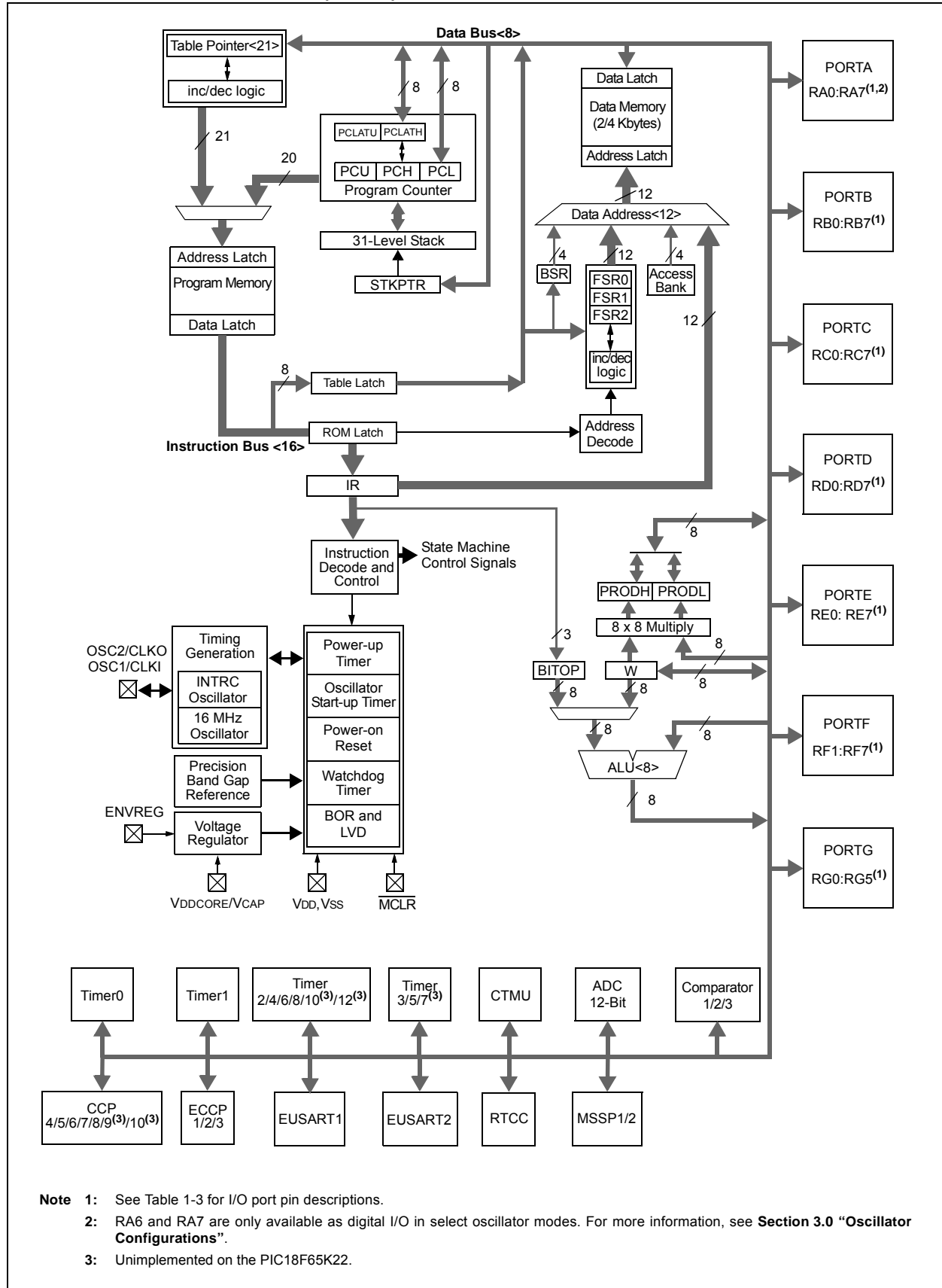
Applications of "[Embedded - Microcontrollers](#)"

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, LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K 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-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f65k22-e-pt

PIC18F87K22 FAMILY

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



PIC18F87K22 FAMILY

TABLE 1-4: PIC18F8XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
RF1/AN6/C2OUT/CTDIN	23			PORTF is a bidirectional I/O port.
RF1		I/O	ST	Digital I/O.
AN6		I	Analog	Analog Input 6.
C2OUT		O	—	Comparator 2 output.
CTDIN		I	ST	CTMU pulse delay input.
RF2/AN7/C1OUT	18			
RF2		I/O	ST	Digital I/O.
AN7		I	Analog	Analog Input 7.
C1OUT		O	—	Comparator 1 output.
RF3/AN8/C2INB/CTMUI	17			
RF3		I/O	ST	Digital I/O.
AN8		I	Analog	Analog Input 8.
C2INB		I	Analog	Comparator 2 Input B.
CTMUI		O	—	CTMU pulse generator charger for the C2INB comparator input.
RF4/AN9/C2INA	16			
RF4		I/O	ST	Digital I/O.
AN9		I	Analog	Analog Input 9.
C2INA		I	Analog	Comparator 2 Input A.
RF5/AN10/C1INB	15			
RF5		I/O	ST	Digital I/O.
AN10		I	Analog	Analog Input 10.
C1INB		I	Analog	Comparator 1 Input B.
RF6/AN11/C1INA	14			
RF6		I/O	ST	Digital I/O.
AN11		I	Analog	Analog Input 11.
C1INA		I	Analog	Comparator 1 Input A.
RF7/AN5/SS1	13			
RF7		I/O	ST	Digital I/O.
AN5		O	Analog	Analog Input 5.
SS1		I	ST	SPI slave select input.

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 VDD)
I²C = I²C™/SMBus

- Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.
2: Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.
3: Not available on PIC18F65K22 and PIC18F85K22 devices.
4: PSP is available only in Microcontroller mode.
5: The CC6, CCP7, CCP8 and CCP9 pin placement depends on the setting of the ECCPMX Configuration bit (CONFIG3H<1>).

6.1.2 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and contained in three separate 8-bit registers.

The low byte, known as the PCL register, is both readable and writable. The high byte, or PCH register, contains the PC<15:8> bits and is not directly readable or writable. Updates to the PCH register are performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits; it is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the Program Counter by any operation that writes PCL. Similarly, the upper two bytes of the Program Counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 6.1.5.1 “Computed GOTO”**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of PCL is fixed to a value of ‘0’. The PC increments by two to address sequential instructions in the program memory.

The CALL, RCALL, GOTO and program branch instructions write to the Program Counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the Program Counter.

6.1.3 RETURN ADDRESS STACK

The return address stack enables execution of any combination of up to 31 program calls and interrupts. The PC is pushed onto the stack when a CALL or RCALL instruction is executed or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction. The value also is pulled off the stack on ADDULNK and SUBULNK instructions, if the extended instruction set is enabled. PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, STKPTR. The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the Top-of-Stack Special Function Registers. Data can also be pushed to, or popped from, the stack using these registers.

A CALL type instruction causes a push onto the stack. The Stack Pointer is first incremented and the location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the CALL). A RETURN type instruction causes a pop from the stack. The contents of the location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

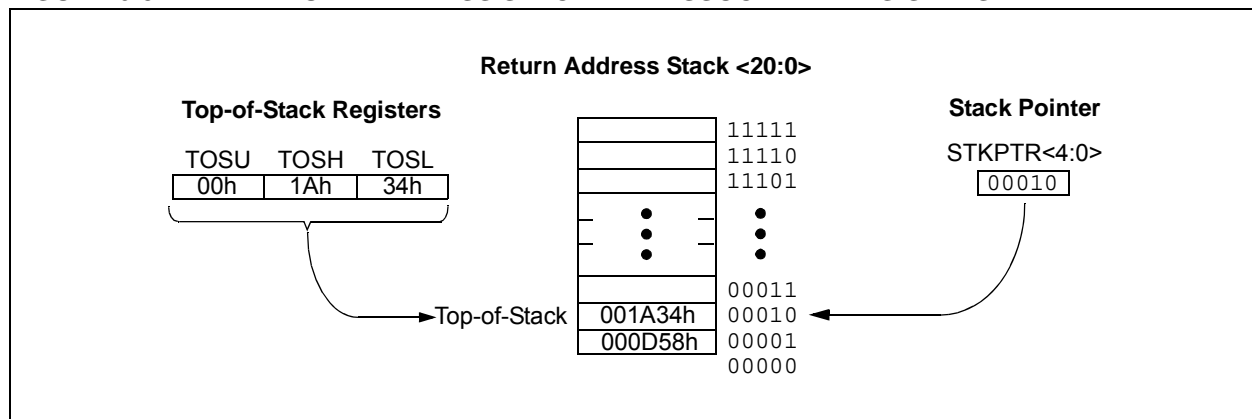
The Stack Pointer is initialized to ‘00000’ after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of ‘00000’; this is only a Reset value. Status bits indicate if the stack is full, has overflowed or has underflowed.

6.1.3.1 Top-of-Stack Access

Only the top of the return address stack (TOS) is readable and writable. A set of three registers, TOSU:TOSH:TOSL, holds the contents of the stack location pointed to by the STKPTR register (Figure 6-3). This allows users to implement a software stack, if necessary. After a CALL, RCALL or interrupt (or ADDULNK and SUBULNK instructions, if the extended instruction set is enabled), the software can read the pushed value by reading the TOSU:TOSH:TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can return these values to TOSU:TOSH:TOSL and do a return.

While accessing the stack, users must disable the Global Interrupt Enable bits to prevent inadvertent stack corruption.

FIGURE 6-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



PIC18F87K22 FAMILY

REGISTER 11-14: PIE5: PERIPHERAL INTERRUPT ENABLE REGISTER 5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR7GIE ⁽¹⁾	TMR12IE ⁽¹⁾	TMR10IE ⁽¹⁾	TMR8IE	TMR7IE ⁽¹⁾	TMR6IE	TMR5IE	TMR4IE
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	TMR7GIE: TMR7 Gate Interrupt Enable bit ⁽¹⁾ 1 = Enabled 0 = Disabled
bit 6	TMR12IE: TMR12 to PR12 Match Interrupt Enable bit ⁽¹⁾ 1 = Enables the TMR12 to PR12 match interrupt 0 = Disables the TMR12 to PR12 match interrupt
bit 5	TMR10IE: TMR10 to PR10 Match Interrupt Enable bit ⁽¹⁾ 1 = Enables the TMR10 to PR10 match interrupt 0 = Disables the TMR10 to PR10 match interrupt
bit 4	TMR8IE: TMR8 to PR8 Match Interrupt Enable bit 1 = Enables the TMR8 to PR8 match interrupt 0 = Disables the TMR8 to PR8 match interrupt
bit 3	TMR7IE: TMR7 Overflow Interrupt Enable bit ⁽¹⁾ 1 = Enables the TMR7 overflow interrupt 0 = Disables the TMR7 overflow interrupt
bit 2	TMR6IE: TMR6 to PR6 Match Interrupt Enable bit 1 = Enables the TMR6 to PR6 match interrupt 0 = Disables the TMR6 to PR6 match interrupt
bit 1	TMR5IE: TMR5 Overflow Interrupt Enable bit 1 = Enables the TMR5 overflow interrupt 0 = Disables the TMR5 overflow interrupt
bit 0	TMR4IE: TMR4 to PR4 Match Interrupt Enable bit 1 = Enables the TMR4 to PR4 match interrupt 0 = Disables the TMR4 to PR4 match interrupt

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

PIC18F87K22 FAMILY

TABLE 11-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

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
INTCON2	RBP \overline{U}	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
PIR1	PSPIP	ADIF	RC1IF	TX1IF	SSP1IF	TMR1GIF	TMR2IF	TMR1IF
PIR2	OSCFIF	—	SSP2IF	BCL2IF	BCL1IF	HLVDIF	TMR3IF	TMR3GIF
PIR3	TMR5GIF	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF
PIR4	CCP10IF ⁽¹⁾	CCP9IF ⁽¹⁾	CCP8IF	CCP7IF	CCP6IF	CCP5IF	CCP4IF	CCP3IF
PIR5	TMR7GIF ⁽¹⁾	TMR12IF ⁽¹⁾	TMR10IF ⁽¹⁾	TMR8IF	TMR7IF ⁽¹⁾	TMR6IF	TMR5IF	TMR4IF
PIR6	—	—	—	EEIF	—	CMP3IF	CMP2IF	CMP1IF
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE
PIE2	OSCFIE	—	SSP2IE	BCL2IE	BCL1IE	HLVDIE	TMR3IE	TMR3GIE
PIE3	TMR5GIE	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE
PIE4	CCP10IE ⁽¹⁾	CCP9IE ⁽¹⁾	CCP8IE	CCP7IE	CCP6IE	CCP5IE	CCP4IE	CCP3IE
PIE5	TMR7GIE ⁽¹⁾	TMR12IE ⁽¹⁾	TMR10IE ⁽¹⁾	TMR8IE	TMR7IE ⁽¹⁾	TMR6IE	TMR5IE	TMR4IE
PIE6	—	—	—	EEIE	—	CMP3IE	CMP2IE	CMP1IE
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP
IPR2	OSCFIP	—	SSP2IP	BCL2IP	BCL1IP	HLVDIP	TMR3IP	TMR3GIP
IPR3	TMR5GIP	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP
IPR4	CCP10IP ⁽¹⁾	CCP9IP ⁽¹⁾	CCP8IP	CCP7IP	CCP6IP	CCP5IP	CCP4IP	CCP3IP
IPR5	TMR7GIP ⁽¹⁾	TMR12IP ⁽¹⁾	TMR10IP ⁽¹⁾	TMR8IP	TMR7IP ⁽¹⁾	TMR6IP	TMR5IP	TMR4IP
IPR6	—	—	—	EEIP	—	CMP3IP	CMP2IP	CMP1IP
RCON	IPEN	SBOREN	\overline{CM}	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}

Legend: Shaded cells are not used by the interrupts.

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

PIC18F87K22 FAMILY

12.4 PORTC, TRISC and LATC Registers

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

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

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

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

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

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

EXAMPLE 12-3: INITIALIZING PORTC

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

TABLE 12-5: PORTC FUNCTIONS

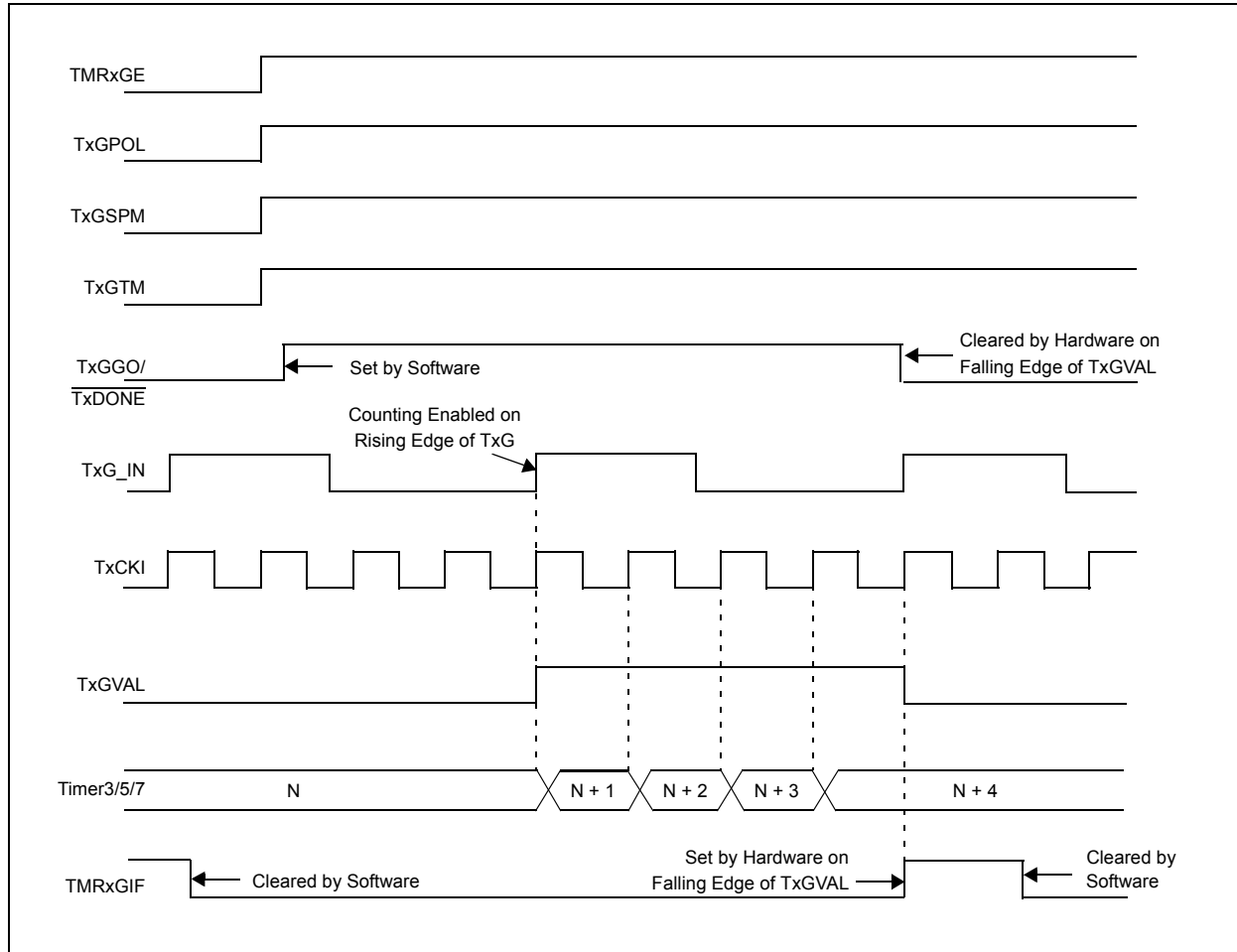
Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RC0/SOSCO/ SCLKI/	RC0	0	O	DIG	LATC<0> data output.
		1	I	ST	PORTC<0> data input.
	SOSCO	1	I	ST	SOSC oscillator output.
	SCLKI	1	I	ST	Digital clock input; enabled when SOSC oscillator is disabled.
RC1/SOSCI/ ECCP2/P2A	RC1	0	O	DIG	LATC<1> data output.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

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, TTL = TTL Buffer Input, I²C = I²C™/SMBus Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

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

PIC18F87K22 FAMILY

FIGURE 16-5: TIMER3/5/7 GATE SINGLE PULSE AND TOGGLE COMBINED MODE



16.5.5 TIMER3/5/7 GATE VALUE STATUS

When Timer3/5/7 gate value status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the TxGVAL bit (TxGCON<2>). The TxGVAL bit is valid even when the Timer3/5/7 gate is not enabled (TMRxGE bit is cleared).

16.5.6 TIMER3/5/7 GATE EVENT INTERRUPT

When the Timer3/5/7 gate event interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of TxGVAL occurs, the TMRxGIF flag bit in the PIRx register will be set. If the TMRxGIE bit in the PIRx register is set, then an interrupt will be recognized.

The TMRxGIF flag bit operates even when the Timer3/5/7 gate is not enabled (TMRxGE bit is cleared).

16.6 Timer3/5/7 Interrupt

The TMRx register pair (TMRxH:TMRxL) increments from 0000h to FFFFh and overflows to 0000h. The Timerx interrupt, if enabled, is generated on overflow and is latched in the interrupt flag bit, TMRxIF. Table 16-3 gives each module's flag bit.

TABLE 16-3: TIMER3/5/7 INTERRUPT FLAG BITS

Timer Module	Flag Bit
3	PIR2<1>
5	PIR5<1>
7	PIR5<3>

This interrupt can be enabled or disabled by setting or clearing the TMRxIE bit, respectively. Table 16-4 gives each module's enable bit.

TABLE 16-4: TIMER3/5/7 INTERRUPT ENABLE BITS

Timer Module	Flag Bit
3	PIE2<1>
5	PIE5<1>
7	PIE5<3>

16.7 Resetting Timer3/5/7 Using the ECCP Special Event Trigger

If the ECCP modules are configured to use Timerx and to generate a Special Event Trigger in Compare mode (CCPxM<3:0> = 1011), this signal will reset Timerx. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled. (For more information, see **Section 20.3.4 "Special Event Trigger"**.)

The module must be configured as either a timer or synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a Period register for Timerx.

If Timerx is running in Asynchronous Counter mode, the Reset operation may not work.

In the event that a write to Timerx coincides with a Special Event Trigger from an ECCP module, the write will take precedence.

Note: The Special Event Triggers from the ECCPx module will only clear the TMR3 register's content, but not set the TMR3IF interrupt flag bit (PIR1<0>).

Note: The CCP and ECCP modules use Timers, 1 through 8, for some modes. The assignment of a particular timer to a CCP/ECCP module is determined by the Timer to CCP enable bits in the CCPTMRSx registers. For more details, see Register 19-2, Register 19-3 and Register 20-2

17.0 TIMER4/6/8/10/12 MODULES

The Timer4/6/8/10/12 timer modules have the following features:

- Eight-bit Timer register (TMRx)
- Eight-bit Period register (PRx)
- Readable and writable (all registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMRx match of PRx

Timer10 and Timer12 are unimplemented for devices with a program memory of 32 Kbytes (PIC18FX5K22).

Note: Throughout this section, generic references are used for register and bit names that are the same, except for an 'x' variable that indicates the item's association with the Timer4, Timer6, Timer8, Timer10 or Timer12 module. For example, the control register is named TxCON and refers to T4CON, T6CON, T8CON, T10CON and T12CON.

The Timer4/6/8/10/12 modules have a control register, which is shown in Register 17-1. Timer4/6/8/10/12 can be shut off by clearing control bit, TMRxON (TxCON<2>), to minimize power consumption. The prescaler and postscaler selection of Timer4/6/8/10/12 are also controlled by this register. Figure 17-1 is a simplified block diagram of the Timer4/6/8/10/12 modules.

17.1 Timer4/6/8/10/12 Operation

Timer4/6/8/10/12 can be used as the PWM time base for the PWM mode of the ECCP modules. The TMRx registers are readable and writable, and are cleared on any device Reset. The input clock (Fosc/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits, TxCKPS<1:0> (TxCON<1:0>). The match output of TMRx goes through a four-bit postscaler (that gives a 1:1 to 1:16 inclusive scaling) to generate a TMRx interrupt, latched in the flag bit, TMRxIF. Table 17-1 shows each module's flag bit.

TABLE 17-1: TIMER4/6/8/10/12 FLAG BITS

Timer Module	Flag Bit PIR5<x>	Timer Module	Flag Bit PIR5<x>
4	0	10	5
6	2	12	6
8	4		

The interrupt can be enabled or disabled by setting or clearing the Timerx Interrupt Enable bit (TMRxIE), shown in Table 17-2.

TABLE 17-2: TIMER4/6/8/10/12 INTERRUPT ENABLE BITS

Timer Module	Flag Bit PIE5<x>	Timer Module	Flag Bit PIE5<x>
4	0	10	5
6	2	12	6
8	4		

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMRx register
- A write to the TxCON register
- Any device Reset – Power-on Reset (POR), MCLR Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR)

A TMRx is not cleared when a TxCON is written.

Note: The CCP and ECCP modules use Timers, 1 through 8, for some modes. The assignment of a particular timer to a CCP/ECCP module is determined by the Timer to CCP enable bits in the CCPTMRSx registers. For more details, see Register 19-2, Register 19-3 and Register 20-2.

PIC18F87K22 FAMILY

REGISTER 18-4: ALRMCFG: ALARM CONFIGURATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
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 **ALRMEN:** Alarm Enable bit
1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00 and CHIME = 0)
0 = Alarm is disabled
- bit 6 **CHIME:** Chime Enable bit
1 = Chime is enabled; ALRMPTR<1:0> bits are allowed to roll over from 00h to FFh
0 = Chime is disabled; ALRMPTR<1:0> bits stop once they reach 00h
- bit 5-2 **AMASK<3:0>:** Alarm Mask Configuration bits
0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every four years)
101x = Reserved – Do not use
11xx = Reserved – Do not use
- bit 1-0 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits
Points to the corresponding Alarm Value registers when reading the ALRMVALH and ALRMVALL registers. The ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.
ALRMVALH:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented
ALRMVALL:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

PIC18F87K22 FAMILY

19.1 CCP Module Configuration

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

19.1.1 CCP MODULES AND TIMER RESOURCES

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

TABLE 19-1: CCP MODE – TIMER RESOURCE

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

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

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

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

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

Note 1: Do not use the reserved bits.

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

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

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

2: Do not use the reserved setting.

PIC18F87K22 FAMILY

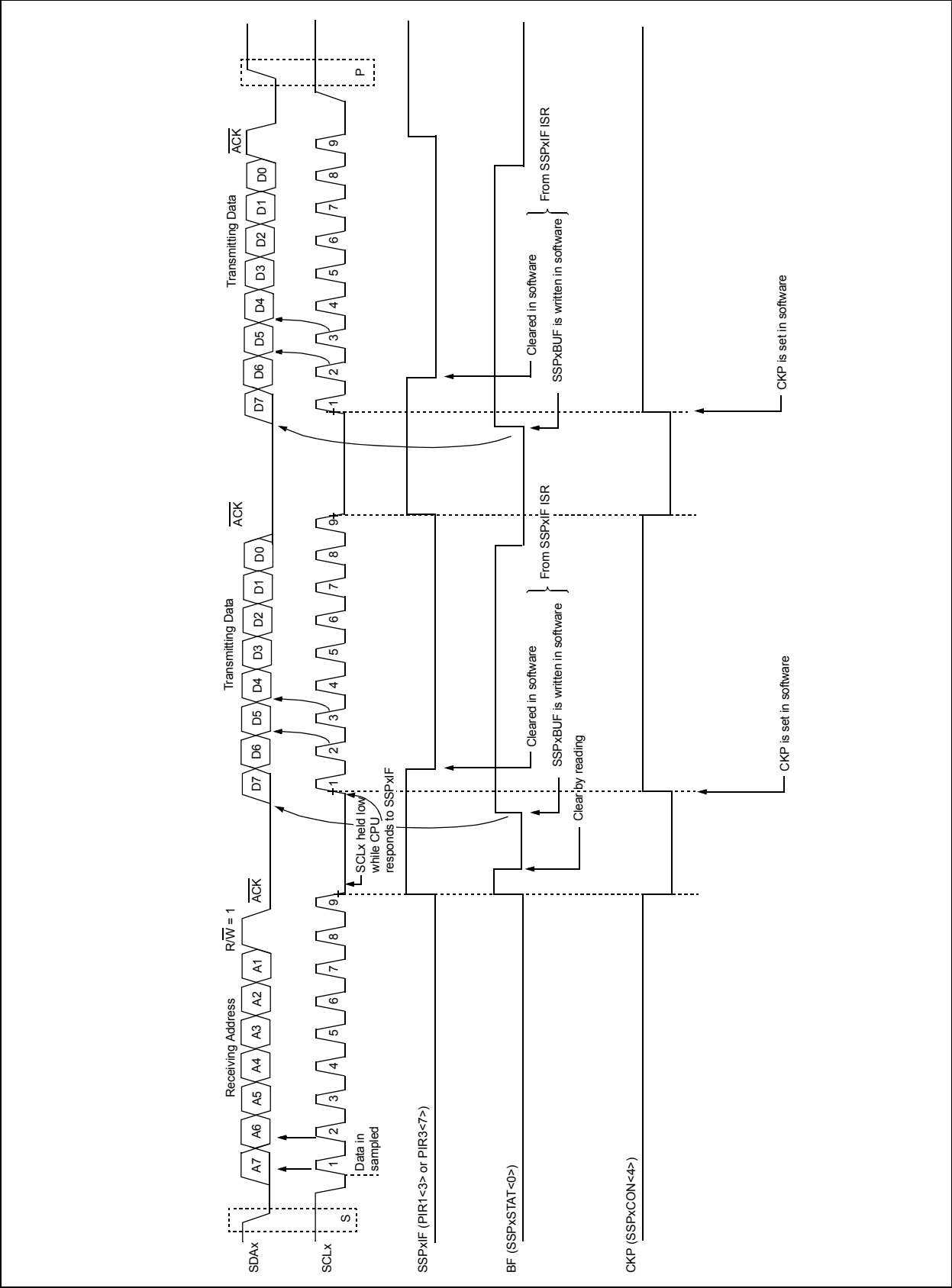
TABLE 21-2: REGISTERS ASSOCIATED WITH SPI OPERATION

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
PIR2	OSCFIF	—	SSP2IF	BCL2IF	BCL1IF	HLVDIF	TMR3IF	TMR3GIF
PIE2	OSCFIE	—	SSP2IE	BCL2IE	BCL1IE	HLVDIE	TMR3IE	TMR3GIE
IPR2	OSCFIP	—	SSP2IP	BCL2IP	BCL1IP	HLVDIP	TMR3IP	TMR3GIP
PIR3	TMR5GIF	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF
PIE3	TMR5GIE	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE
IPR3	TMR5GIP	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—
SSP1BUF	MSSP1 Receive Buffer/Transmit Register							
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
SSP1STAT	SMP	CKE	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
SSP2STAT	SMP	CKE	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
SSP2BUF	MSSP2 Receive Buffer/Transmit Register							
ODCON1	SSP1OD	CCP2OD	CCP1OD	—	—	—	—	SSP2OD
PMD0	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSP2MD	SSP1MD	ADCMD

Legend: Shaded cells are not used by the MSSP module in SPI mode.

PIC18F87K22 FAMILY

FIGURE 21-10: I²C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)



22.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex, asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

All members of the PIC18F87K22 family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-wake-up on character reception
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1) and PORTG (RG1/TX2/CK2/AN19/C3OUT and RG2/RX2/DT2/AN18/C3INA), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
 - Bit, SPEN (RCSTA1<7>), must be set (= 1)
 - Bit, TRISC<7>, must be set (= 1)
 - Bit, TRISC<6>, must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - Bit, TRISC<6>, must be set (= 1) for Synchronous Slave mode
- For EUSART2:
 - Bit, SPEN (RCSTA2<7>), must be set (= 1)
 - Bit, TRISG<2>, must be set (= 1)
 - Bit TRISG<1> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - Bit, TRISC<6>, must be set (= 1) for Synchronous Slave mode

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are detailed on the following pages in Register 22-1, Register 22-2 and Register 22-3, respectively.

Note: Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

PIC18F87K22 FAMILY

FIGURE 22-7: ASYNCHRONOUS RECEPTION

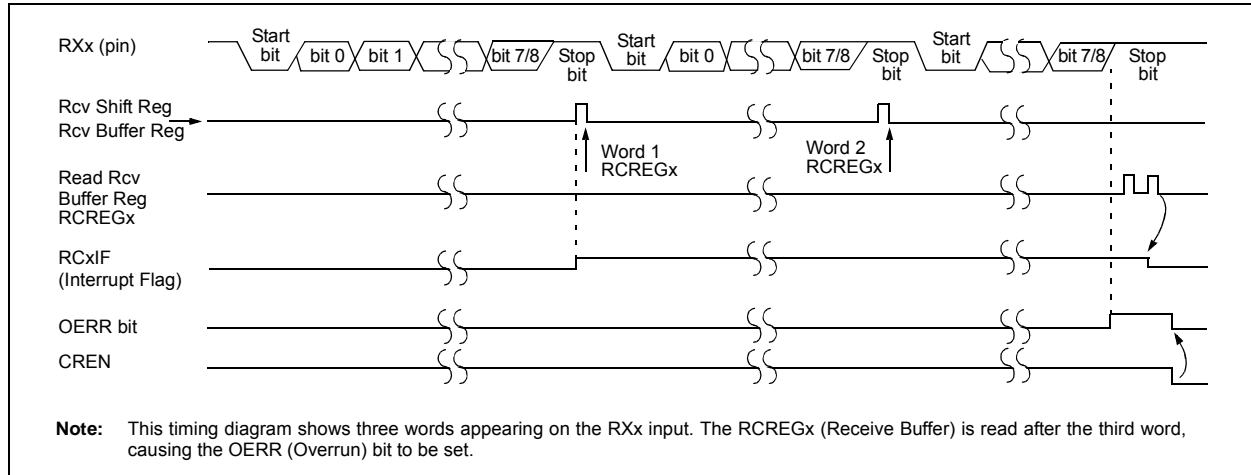


TABLE 22-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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
PIR3	TMR5GIF	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF
PIE3	TMR5GIE	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE
IPR3	TMR5GIP	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
RCREG1	EUSART1 Receive Register							
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH1	EUSART1 Baud Rate Generator Register High Byte							
SPBRG1	EUSART1 Baud Rate Generator Register							
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
RCREG2	EUSART2 Receive Register							
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH2	EUSART2 Baud Rate Generator Register High Byte							
SPBRG2	EUSART2 Baud Rate Generator Register							
ODCON3	U2OD	U1OD	—	—	—	—	—	CTMUDS
PMD0	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSP2MD	SSP1MD	ADCMD

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

27.7 Creating a Delay with the CTMU Module

A unique feature on board the CTMU module is its ability to generate system clock independent output pulses, based on either an external voltage or an external capacitor value. When using an external voltage, this is accomplished using the CTDIN input pin as a trigger for the pulse delay. When using an external capacitor value, this is accomplished using the internal comparator voltage reference module and Comparator 2 input pin. The pulse is output onto the CTPLS pin. To enable this mode, set the TGEN bit.

See Figure 27-4 for an example circuit. When CTMUDS (ODCON3<0>) is cleared, the pulse delay is determined by the output of Comparator 2, and when it is set, the pulse delay is determined by the input of CTDIN. CDELAY is chosen by the user to determine the output pulse width on CTPLS. The pulse width is calculated by $T = (C_{DELAY}/I) * V$, where I is known from the current source measurement step (Section 27.4.1 “Current Source Calibration”) and V is the Internal Reference Voltage (CVREF).

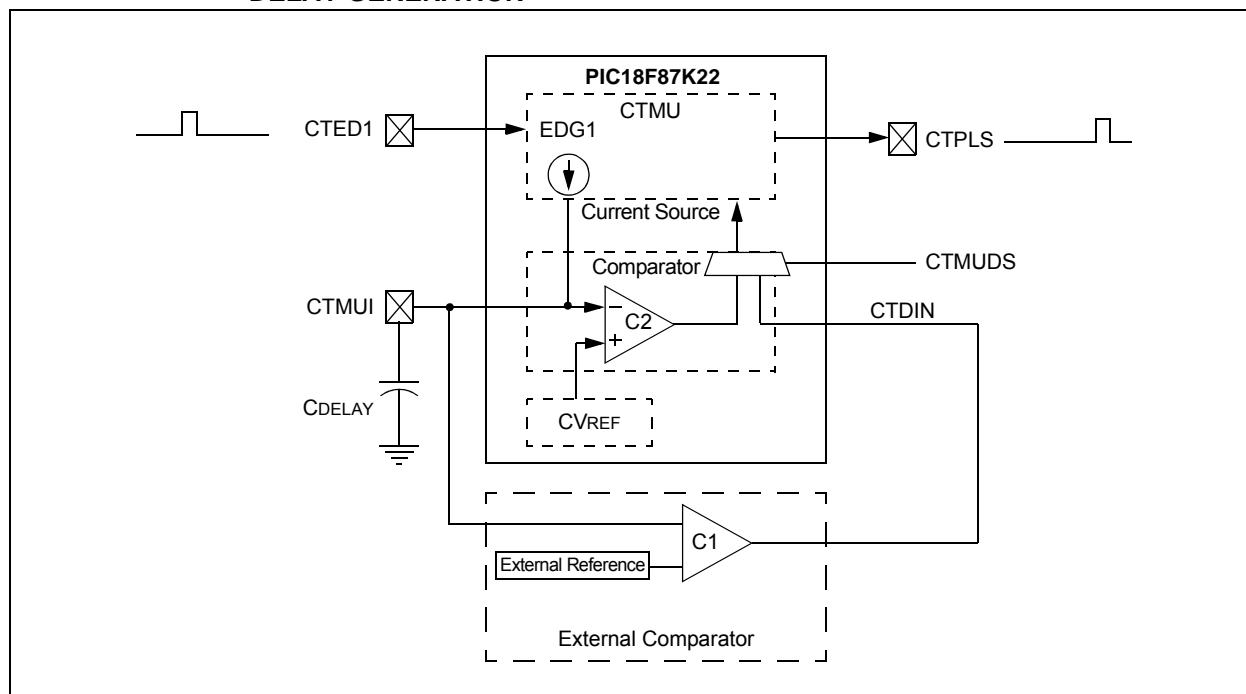
An example use of the external capacitor feature is interfacing with variable capacitive-based sensors, such as a humidity sensor. As the humidity varies, the pulse-width output on CTPLS will vary. An example use of the CTDIN feature is interfacing with a digital sensor. The CTPLS output pin can be connected to an input capture pin and the varying pulse width measured to determine the sensor's output in the application.

To use this feature:

1. If CTMUDS is cleared, initialize Comparator 2.
2. If CTMUDS is cleared, initialize the comparator voltage reference.
3. Initialize the CTMU and enable time delay generation by setting the TGEN bit.
4. Set EDG1STAT.

When CTMUDS is cleared, as soon as CDELAY charges to the value of the voltage reference trip point, an output pulse is generated on CTPLS. When CTMUDS is set, as soon as CTDIN is set, an output pulse is generated on CTPLS.

FIGURE 27-4: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



PIC18F87K22 FAMILY

MULLW Multiply Literal with W

Syntax:	MULLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	(W) \times k \rightarrow PRODH:PRODL			
Status Affected:	None			
Encoding:	0000	1101	kkkk	kkkk
Description:	An unsigned multiplication is carried			

Description: An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in the PRODH:PRODL register pair. PRODH contains the high byte.

W is unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL

Example: MULLW 0C4h

Before Instruction

W = E2h
PRODH = ?
PRODL = ?

After Instruction

W = E2h
PRODH = ADh
PRODL = 08h

MULWF Multiply W with f

Syntax:	MULWF f{,a}			
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$			
Operation:	$(W) \times (f) \rightarrow \text{PRODH:PRODL}$			
Status Affected:	None			
Encoding:	0000	001a	ffff	ffff

Description: An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

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 registers PRODH: PRODL

Example: MULWF REG, 1

Before Instruction

W = C4h
REG = B5h
PRODH = ?
PRODL = ?

After Instruction

W = C4h
REG = B5h
PRODH = 8Ah
PRODL = 94h

PIC18F87K22 FAMILY

29.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set for the PIC18F87K22 family. This includes the MPLAB C18 C Compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option or dialog box within the environment that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.

30.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

30.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

30.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

30.10 PICkit 3 In-Circuit Debugger/Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

FIGURE 31-15: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)

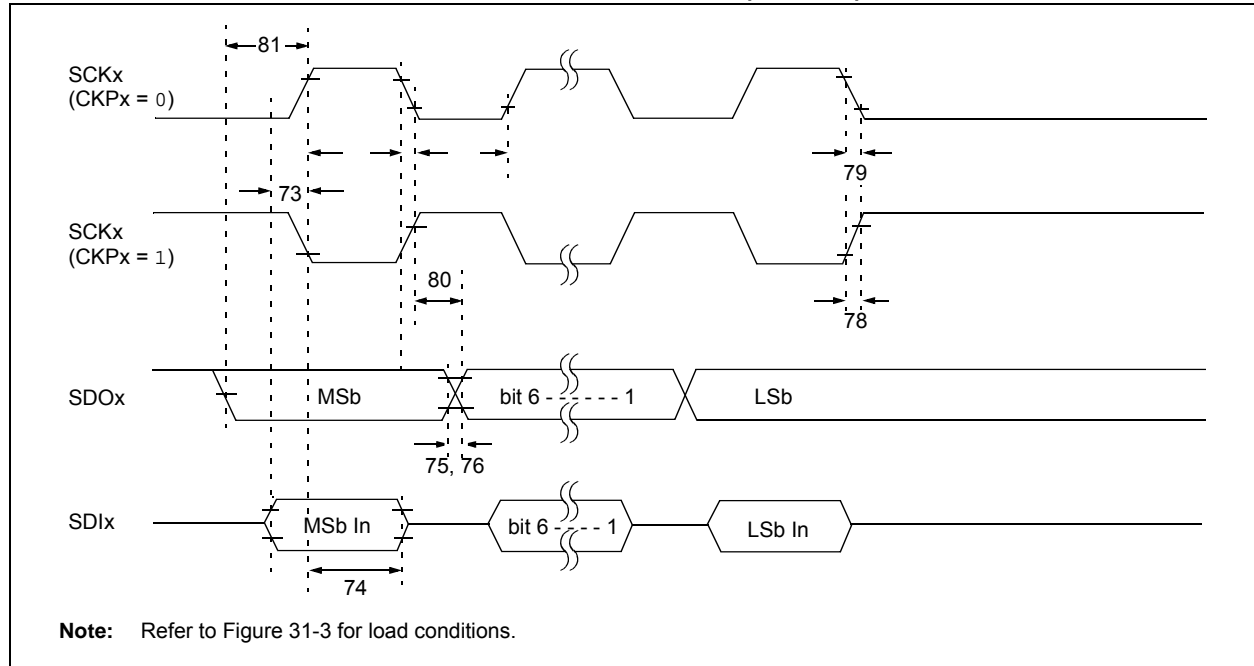


TABLE 31-18: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TdIV2scH, TdIV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	ns	
73A	Tb2B	Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2	1.5 Tcy + 40	—	ns	
74	Tsch2dIL, TscL2dIL	Hold Time of SDIx Data Input to SCKx Edge	40	—	ns	
75	TdoR	SDOx Data Output Rise Time	—	25	ns	
76	TdoF	SDOx Data Output Fall Time	—	25	ns	
78	TscR	SCKx Output Rise Time (Master mode)	—	25	ns	
79	TscF	SCKx Output Fall Time (Master mode)	—	25	ns	
80	Tsch2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	50	ns	
81	TdoV2sch, TdoV2scL	SDOx Data Output Setup to SCKx Edge	Tcy	—	ns	