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	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f25k80-i-ss

4.6 Exiting Idle and Sleep Modes

An exit from Sleep mode or any of the Idle modes is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power-managed modes. The clocking subsystem actions are discussed in each of the power-managed modes (see **Section 4.2 “Run Modes”, Section 4.3 “Sleep Mode”** and **Section 4.4 “Idle Modes”**).

4.6.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode or Sleep mode to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCONx or PIEx registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

On all exits from Idle or Sleep modes by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 10.0 “Interrupts”**).

4.6.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power-managed mode (see **Section 4.2 “Run Modes”** and **Section 4.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 28.2 “Watchdog Timer (WDT)”**).

Executing a `SLEEP` or `CLRWDT` instruction clears the WDT timer and postscaler, loses the currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifies the IRCFx bits in the OSCCON register (if the internal oscillator block is the device clock source).

4.6.3 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock becomes ready. At that time, the OSTS bit is set and the device begins executing code. If the internal oscillator block is the new clock source, the HFIOFS/MFIOFS bits are set instead.

The exit delay time from Reset to the start of code execution depends on both the clock sources before and after the wake-up, and the type of oscillator, if the new clock source is the primary clock. Exit delays are summarized in Table 4-4.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see **Section 28.4 “Two-Speed Start-up”**) or Fail-Safe Clock Monitor (see **Section 28.5 “Fail-Safe Clock Monitor”**) is enabled, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Execution is clocked by the internal oscillator block until either the primary clock becomes ready or a power-managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

4.6.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. The two cases are:

- When in PRI_IDLE mode, where the primary clock source is not stopped
- When the primary clock source is not any of the LP, XT, HS or HSPLL modes

In these instances, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally, does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval, T_{CSD}, following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

TABLE 5-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
BAUDCON2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	01x0 0-00	01x0 0-00	uuuu u-uu
IPR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 -111	1111 -111	uuuu -uuu
PIR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 -000	0000 -000	uuuu -uuu
PIE4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 -000	0000 -000	uuuu -uuu
CVRCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xx-- ----	xx-- ----	uu-- ----
TMR3H	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
T3GCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0x00	0000 0x00	uuuu u-uu
SPBRG1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
TXSTA1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 000x	0000 000x	uuuu uuuu
T1GCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0x00	0000 0x00	uuuu u-uu
PR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
HLVDCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	01x0 0-00	01x0 0-00	uuuu u-uu
RCSTA2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 000x	0000 000x	uuuu uuuu
IPR3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	--11 111-	--11 111-	--uu uuu-
PIR3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	--00 000-	--x0 xxx-	--uu uuu-
PIE3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	--00 000-	0000 0000	uuuu uuuu
IPR2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1--- 1111	1--- 111x	u--- uuuu
PIR2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0--- 0000	0--- 000x	u--- uuuu(1)
PIE2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0--- 0000	0--- 0000	u--- uuuu
IPR1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-111 1111	-111 1111	-uuu uuuu
PIR1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu(1)
	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-000 0000	-000 0000	-uuu uuuu
PIE1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-000 0000	-000 0000	-uuu uuuu
PSTR1CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	00-0 0001	xx-x xxxx	—
OSCTUNE	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
REFOCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0-00 0000	0-00 0000	u-uu uuuu

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

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: 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.

4: See Table 5-3 for Reset value for specific conditions.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

6.4.3.1 FSR Registers and the INDF Operand

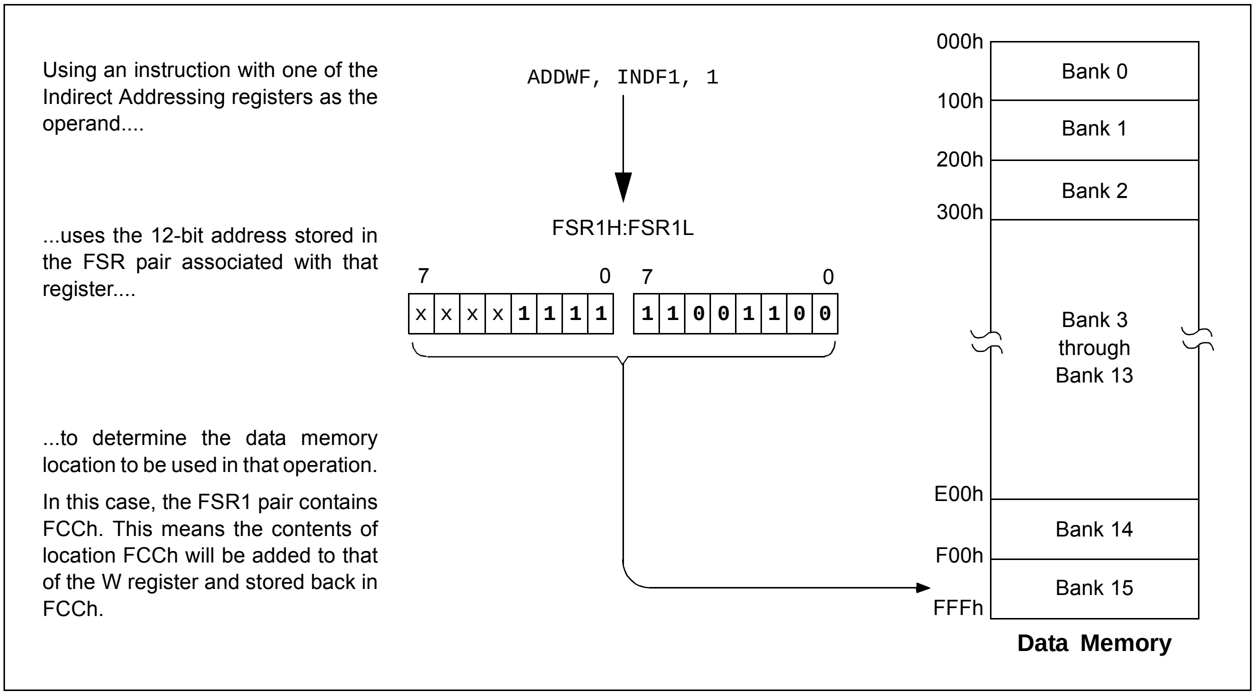
At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers: FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of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

Indirect Addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as “virtual” registers. The operands are mapped in the SFR space, but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L.

Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction’s target. The INDF operand is just a convenient way of using the pointer.

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

FIGURE 6-8: INDIRECT ADDRESSING



11.2 PORTA, TRISA and LATA Registers

PORTA is a seven-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISA and LATA.

RA5 and RA<3:0> are multiplexed with analog inputs for the A/D Converter.

The operation of the analog inputs as A/D Converter inputs is selected by clearing or setting the ANSELx control bits in the ANCON1 register. The corresponding TRISA bits control the direction of these pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

Note:	RA5 and RA<3:0> are configured as analog inputs on any Reset and are read as '0'.

OSC2/CLKO/RA6 and OSC1/CLKI/RA7 normally serve as the external circuit connections for the external (primary) oscillator circuit (HS Oscillator modes) or the external clock input and output (EC Oscillator modes). In these cases, RA6 and RA7 are not available as digital I/O and their corresponding TRIS and LAT bits are read as '0'. When the device is configured to use HF-INTOSC, MF-INTOSC or LF-INTOSC as the default oscillator mode, RA6 and RA7 are automatically configured as digital I/O; the oscillator and clock in/clock out functions are disabled.

RA5 has additional functionality for Timer1 and Timer3. It can be configured as the Timer1 clock input or the Timer3 external clock gate input.

EXAMPLE 11-1: INITIALIZING PORTA

```
CLRF   PORTA   ; Initialize PORTA by
               ; clearing output latches
CLRF   LATA    ; Alternate method to
               ; clear output data latches
MOVLW  00h     ; Configure A/D
MOVWF  ANCON1  ; for digital inputs
MOVLW  0BFh    ; Value used to initialize
               ; data direction
MOVWF  TRISA   ; Set RA<7, 5:0> as inputs,
               ; RA<6> as output
```

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
ODCON	SSPOD	CCP5OD	CCP4OD	CCP3OD	CCP2OD	CCP1OD	U2OD	U1OD
ANCON1	—	ANSEL14	ANSEL13	ANSEL12	ANSEL11	ANSEL10	ANSEL9	ANSEL8

Legend: Shaded cells are not used by PORTB.

TABLE 11-7: PORTD FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD7/RX2/DT2/ P1D/PSP7	RD7	0	O	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	RX2(1)	1	I	ST	Asynchronous serial receive data input (EUSARTx module).
	DT2(1)	1	O	DIG	Synchronous serial data output (EUSARTx module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSARTx module); user must configure as an input.
	P1D	0	O	DIG	ECCP1 Enhanced PWM output, Channel D. May be configured for tri-state during Enhanced PWM.
	PSP7	x	I/O	ST	Parallel Slave Port data.

Legend: O = Output; I = Input; ANA = Analog Signal; DIG = CMOS Output; ST = Schmitt Trigger Buffer Input;
x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: This is the pin assignment for 40 and 44-pin devices (PIC18F4XK80).

TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
PADCFG1	RDPU(1)	REPU(1)	RFPU(2)	RGPU(2)	—	—	—	CTMUDS
ODCON	SSPOD	CCP5OD	CCP4OD	CCP3OD	CCP2OD	CCP1OD	U2OD	U1OD
ANCON1	—	ANSEL14	ANSEL13	ANSEL12	ANSEL11	ANSEL10	ANSEL9	ANSEL8

Legend: Shaded cells are not used by PORTD.

Note 1: These bits are unimplemented on 28-pin devices, read as '0'.

2: These bits are unimplemented on 28/40/44-pin devices, read as '0'.

FIGURE 12-4: CARRIER LOW SYNCHRONIZATION (MDCHSYNC = 0, MDCLSYNC = 1)

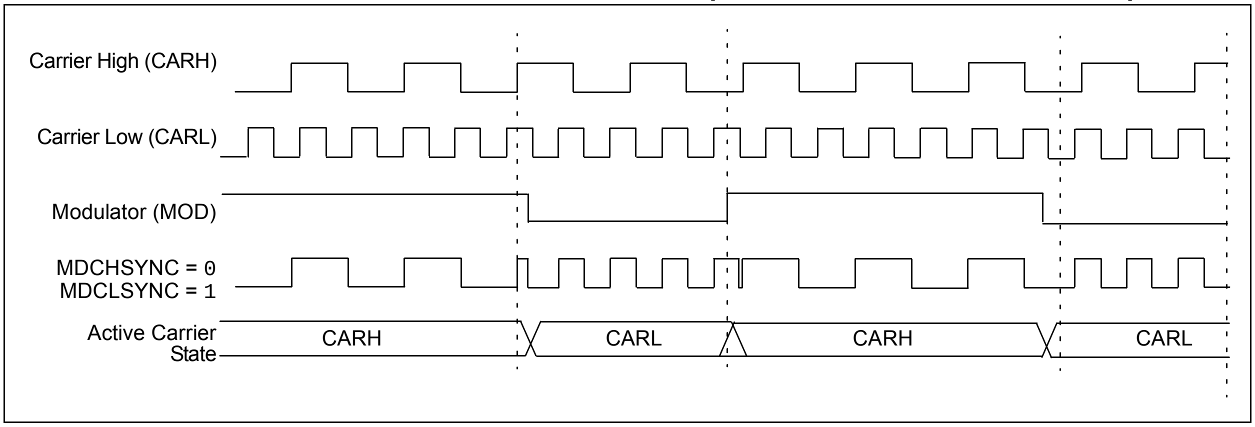


FIGURE 12-5: FULL SYNCHRONIZATION (MDCHSYNC = 1, MDCLSYNC = 1)

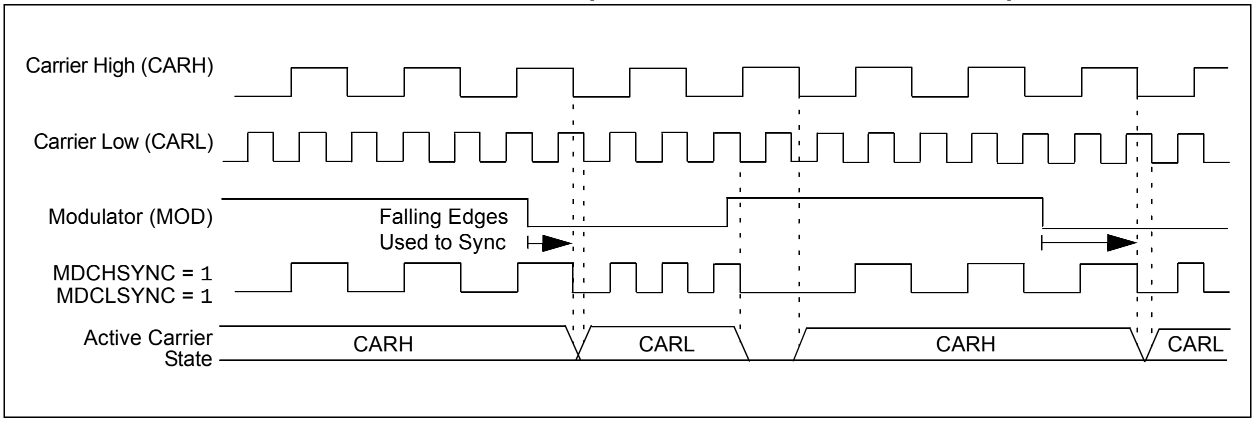
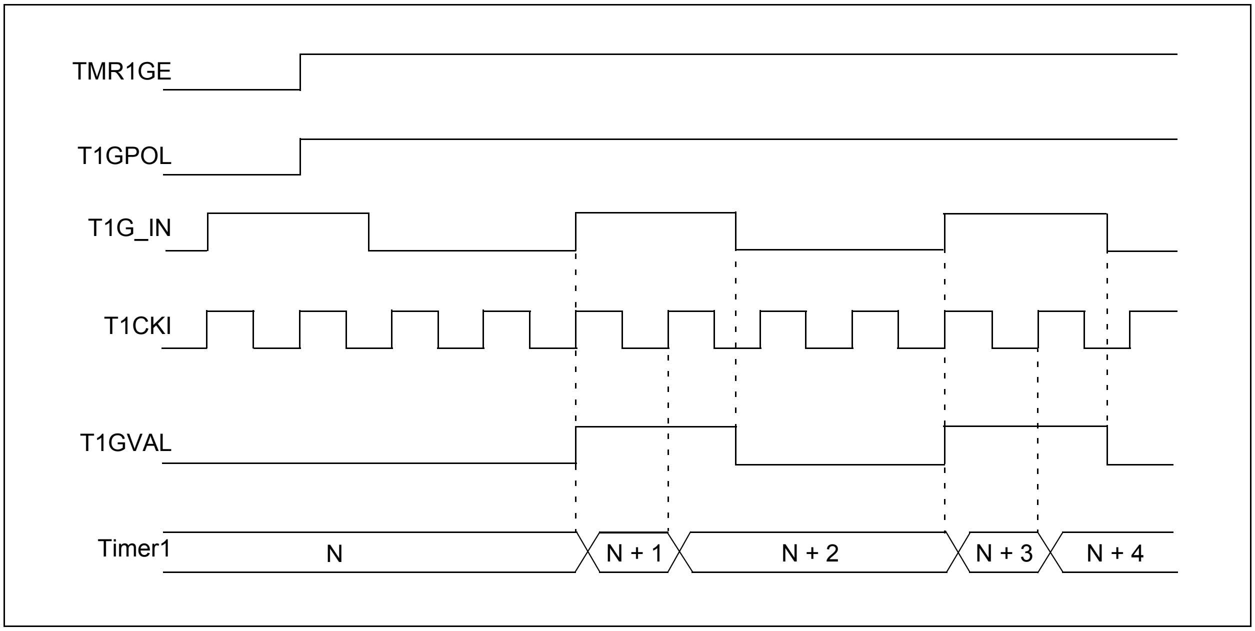


FIGURE 14-4: TIMER1 GATE COUNT ENABLE MODE



14.8.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four sources. Source selection is controlled by the T1GSSx (T1GCON<1:0>) bits (see Table 14-4).

TABLE 14-4: TIMER1 GATE SOURCES

T1GSS<1:0>	Timer1 Gate Source
00	Timer1 Gate Pin
01	TMR2 to Match PR2 (TMR2 increments to match PR2)
10	Comparator 1 Output (comparator logic high output)
11	Comparator 2 Output (comparator logic high output)

The polarity for each available source is also selectable, controlled by the T1GPOL bit (T1GCON<6>).

14.8.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

14.8.2.2 Timer2 Match Gate Operation

The TMR2 register will increment until it matches the value in the PR2 register. On the very next increment cycle, TMR2 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry. The pulse will remain high for one instruction cycle and will return back to a low state until the next match.

Depending on T1GPOL, Timer1 increments differently when TMR2 matches PR2. When T1GPOL = 1, Timer1 increments for a single instruction cycle following a TMR2 match with PR2. When T1GPOL = 0, Timer1 increments continuously except for the cycle following the match when the gate signal goes from low-to-high.

14.8.2.3 Comparator 1 Output Gate Operation

The output of Comparator 1 can be internally supplied to the Timer1 gate circuitry. After setting up Comparator 1 with the CM1CON register, Timer1 will increment depending on the transitions of the CMP1OUT (CMSTAT<6>) bit.

14.8.2.4 Comparator 2 Output Gate Operation

The output of Comparator 2 can be internally supplied to the Timer1 gate circuitry. After setting up Comparator 2 with the CM2CON register, Timer1 will increment depending on the transitions of the CMP2OUT (CMSTAT<7>) bit.

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

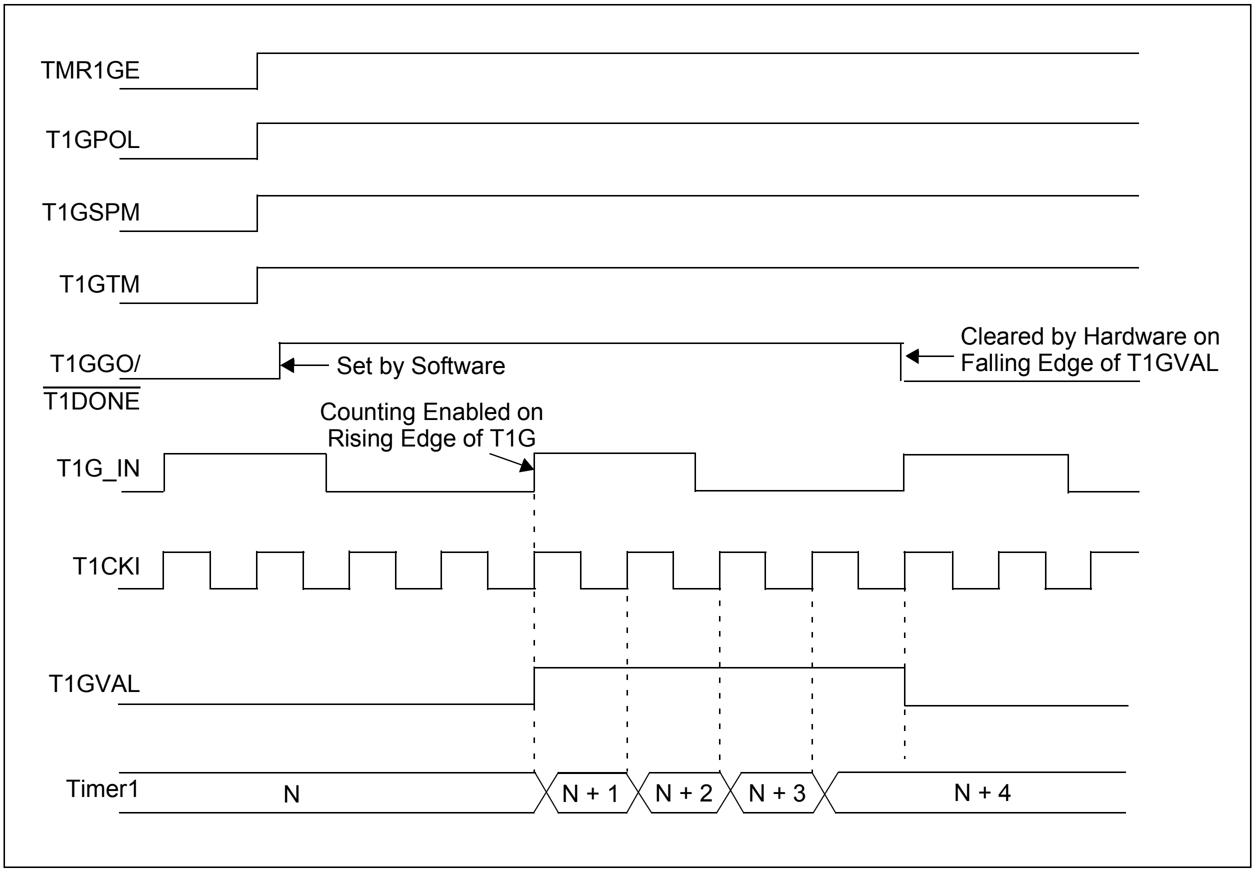


TABLE 14-5: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	TMR1GIF	TMR2IF	TMR1IF
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	TMR1GIE	TMR2IE	TMR1IE
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	TMR1GIP	TMR2IP	TMR1IP
TMR1L	Timer1 Register Low Byte							
TMR1H	Timer1 Register High Byte							
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	SOSCEN	$\overline{\text{T1SYNC}}$	RD16	TMR1ON
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{T1DONE}}$	T1GVAL	T1GSS1	T1GSS0
OSCCON2	—	SOSCRUN	—	SOSCDRV	SOSCGO	—	MFIOFS	MFIOSEL
PMD1	PSPMD	CTMUMD	ADCMD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD

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

TABLE 19-5: REGISTERS ASSOCIATED WITH PWM AND TIMERS

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
RCON	IPEN	SBOREN	$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
PIR3	—	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	—
PIE3	—	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	—
IPR3	—	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	—
PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	—	CCP5IF	CCP4IF	CCP3IF
PIE4	TMR4IE	EEIE	CMP2IE	CMP1IE	—	CCP5IE	CCP4IE	CCP3IE
IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP	—	CCP5IP	CCP4IP	CCP3IP
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
TMR2	Timer2 Register							
TMR4	Timer4 Register							
PR2	Timer2 Period Register							
PR4	Timer4 Period Register							
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
CCPR2L	Capture/Compare/PWM Register 2 Low Byte							
CCPR2H	Capture/Compare/PWM Register 2 High Byte							
CCPR3L	Capture/Compare/PWM Register 3 Low Byte							
CCPR3H	Capture/Compare/PWM Register 3 High Byte							
CCPR4L	Capture/Compare/PWM Register 4 Low Byte							
CCPR4H	Capture/Compare/PWM Register 4 High Byte							
CCPR5L	Capture/Compare/PWM Register 5 Low Byte							
CCPR5H	Capture/Compare/PWM Register 5 High Byte							
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
CCP3CON	—	—	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0
CCPTMRS	—	—	—	C5TSEL	C4TSEL	C3TSEL	C2TSEL	C1TSEL
PMD0	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSPMD

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2/4.

REGISTER 20-4: ECCP1DEL: ENHANCED PWM CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0
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 **P1RSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCP1ASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCP1ASE must be cleared by software to restart the PWM

bit 6-0 **P1DC<6:0>:** PWM Delay Count bits

P1DCn = Number of FOSC/4 (4 * TOSC) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it does transition active.

20.4.7 PULSE STEERING MODE

In Single Output mode, pulse steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can simultaneously be available on multiple pins.

Once the Single Output mode is selected (CCP1M<3:2> = 11 and P1M<1:0> = 00 of the CCP1CON register), the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STR<D:A> bits (PSTR1CON<3:0>), as provided in Table 20-2.

> **Note:** The associated TRIS bits must be set to output (‘0’) to enable the pin output driver in order to see the PWM signal on the pin.

While the PWM Steering mode is active, the CCP1M<1:0> bits (CCP1CON<1:0>) select the PWM output polarity for the P1<D:A> pins.

The PWM auto-shutdown operation also applies to the PWM Steering mode, as described in **Section 20.4.4 “Enhanced PWM Auto-shutdown mode”**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

21.4.3.4 7-Bit Address Masking Mode

Unlike 5-bit masking, 7-Bit Address Masking mode uses a mask of up to 8 bits (in 10-bit addressing) to define a range of addresses that can be Acknowledged, using the lowest bits of the incoming address. This allows the module to Acknowledge up to 127 different addresses with 7-bit addressing, or 255 with 10-bit addressing (see Example 21-3). This mode is the default configuration of the module, which is selected when MSSPMSK is unprogrammed (‘1’).

The address mask for 7-Bit Address Masking mode is stored in the SSPMSK register, instead of the SSPCON2 register. SSPMSK is a separate hardware register within the module, but it is not directly addressable. Instead, it shares an address in the SFR space with the SSPADD register. To access the SSPMSK register, it is necessary to select MSSP mode, ‘1001’ (SSPCON1<3:0> = 1001) and then read or write to the location of SSPADD.

To use 7-Bit Address Masking mode, it is necessary to initialize SSPMSK with a value before selecting the I^{2}C Slave Addressing mode. Thus, the required sequence of events is:

1. Select SSPMSK Access mode (SSPCON2<3:0> = 1001).
2. Write the mask value to the appropriate SSPADD register address (FC8h).
3. Set the appropriate I^{2}C Slave mode (SSPCON2<3:0> = 0111 for 10-bit addressing, 0110 for 7-bit addressing).

Setting or clearing mask bits in SSPMSK behaves in the opposite manner of the ADMSKx bits in 5-Bit Address Masking mode. That is, clearing a bit in SSPMSK causes the corresponding address bit to be masked; setting the bit requires a match in that position. SSPMSK resets to all ‘1’s upon any Reset condition and, therefore, has no effect on the standard MSSP operation until written with a mask value.

With 7-bit addressing, SSPMSK<7:1> bits mask the corresponding address bits in the SSPADD register. For any SSPMSK bits that are active (SSPMSK<n> = 0), the corresponding SSPADD address bit is ignored (SSPADD<n> = x). For the module to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

With 10-bit addressing, SSPMSK<7:0> bits mask the corresponding address bits in the SSPADD register. For any SSPMSK bits that are active (= 0), the corresponding SSPADD address bit is ignored (SSPADD<n> = x).

Note:	The two Most Significant bits of the address are not affected by address masking.

EXAMPLE 21-3: ADDRESS MASKING EXAMPLES IN 7-BIT MASKING MODE

7-Bit Addressing:

SSPADD<7:1> = 1010 000

SSPMSK<7:1> = 1111 001

Addresses Acknowledged = ACh, A8h, A4h, A0h

10-Bit Addressing:

SSPADD<7:0> = 1010 0000 (The two MSb are ignored in this example since they are not affected)

SSPMSK<5:1> = 1111 0011

Addresses Acknowledged = ACh, A8h, A4h, A0h

21.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^{2}C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I^{2}C protocol. It consists of all '0's with R/$\overline{\text{W}}$ = 0.

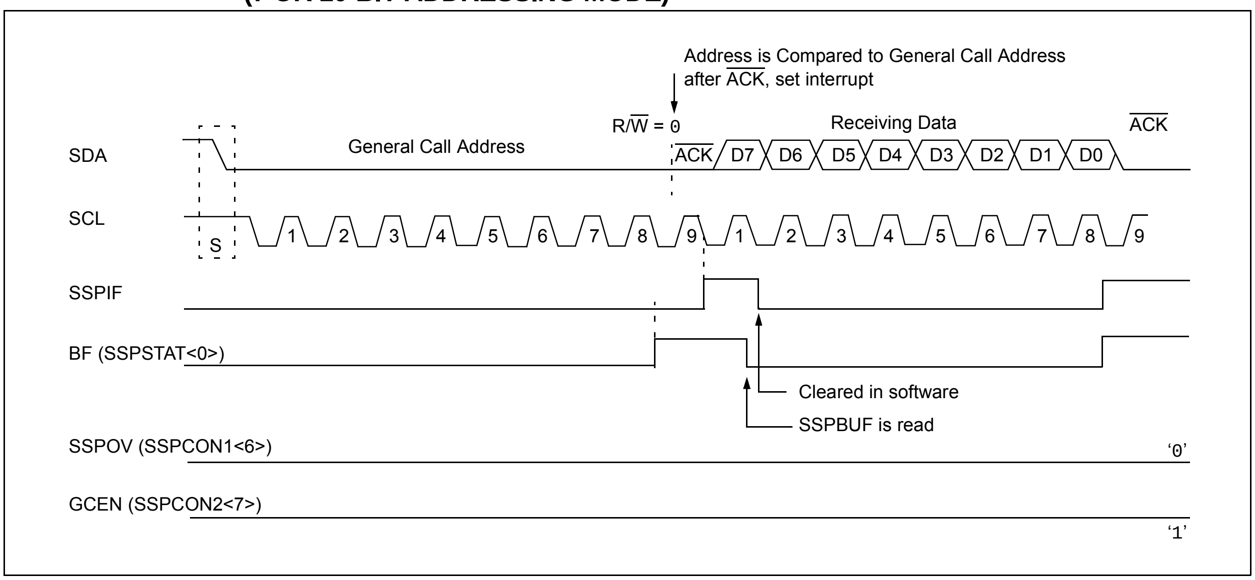
The general call address is recognized when the General Call Enable bit, GCEN, is enabled (SSPCON2<7> set). Following a Start bit detect, eight bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit), and on the falling edge of the ninth bit ($\overline{\text{ACK}}$ bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device-specific or a general call address.

In 10-Bit Addressing mode, the SSPADD is required to be updated for the second half of the address to match and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Addressing mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 21-17).

FIGURE 21-17: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESSING MODE)



EXAMPLE 22-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, and 8-bit BRG:

Desired Baud Rate = FOSC/(64 ([SPBRGHx:SPBRGx] + 1))

Solving for SPBRGHx:SPBRGx:

X = ((FOSC/Desired Baud Rate)/64) – 1
= ((16000000/9600)/64) – 1
= [25.042] = 25

Calculated Baud Rate = 16000000/(64 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
= (9615 – 9600)/9600 = 0.16%

TABLE 22-3: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH1	EUSART1 Baud Rate Generator Register High Byte							
SPBRG1	EUSART1 Baud Rate Generator Register							
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH2	EUSART2 Baud Rate Generator Register High Byte							
SPBRG2	EUSART2 Baud Rate Generator Register Low Byte							
PMD0	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSPMD

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the BRG.

22.3.2 EUSARTx ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 22-6. The data is received on the RXx pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

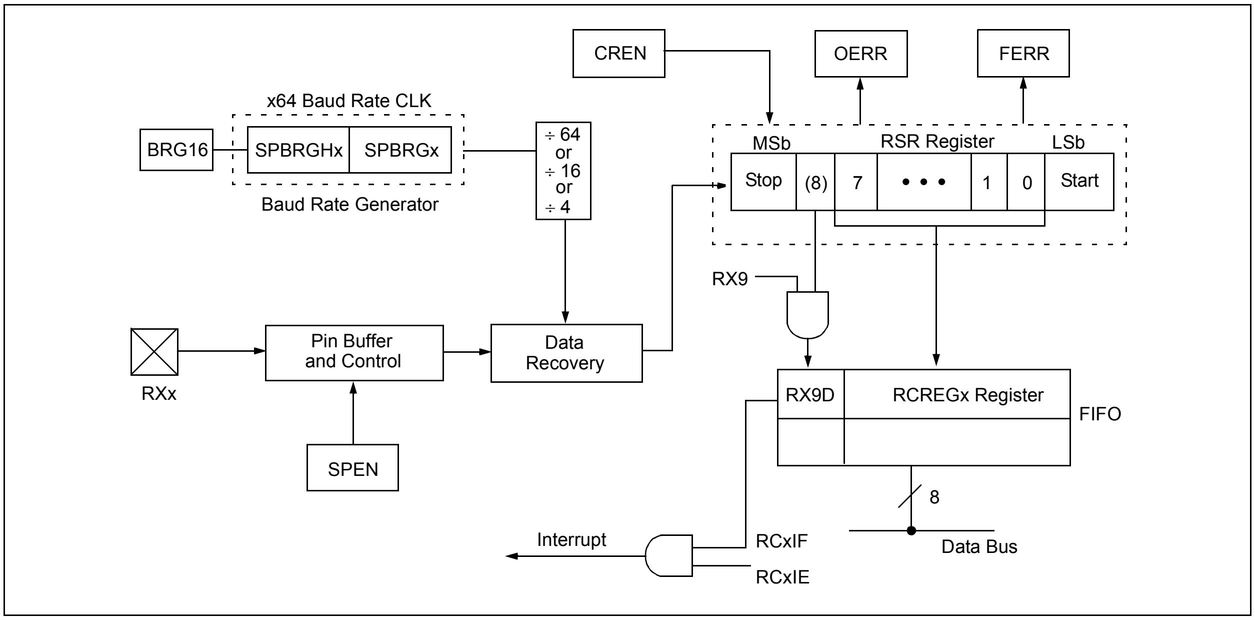
1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RCxIE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RCxIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCxIE, was set.
7. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREGx register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits (INTCON<7:6>) are set.

22.3.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCxIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCxIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCxIE and GIE bits are set.
8. Read the RCSTAx register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREGx to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 22-6: EUSARTx RECEIVE BLOCK DIAGRAM



25.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 32-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

A block diagram of the module is shown in Figure 25-1. The resistor ladder is segmented to provide a range of CVREF values and has a power-down function to conserve power when the reference is not being used. The module's supply reference can be provided from either device VDD/VSS or an external voltage reference.

25.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register (Register 25-1). The comparator voltage reference provides a range of output voltage with 32 levels.

The CVR<4:0> selection bits (CVRCON<4:0>) offer a range of output voltages. Equation 25-1 shows the how the comparator voltage reference is computed.

EQUATION 25-1:

If CVRSS = 1:

$$CV_{REF} = \left(V_{REF-} + \frac{CVR<4:0>}{32}\right) \bullet (V_{REF+} - V_{REF-})$$

If CVRSS = 0:

$$CV_{REF} = \left(AV_{SS} + \frac{CVR<4:0>}{32}\right) \bullet (AV_{DD} - AV_{SS})$$

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF- that are multiplexed with RA3 and RA2. The voltage source is selected by the CVRSS bit (CVRCON<5>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 31-2 in **Section 31.0 "Electrical Characteristics"**).

REGISTER 25-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0
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 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit
1 = CVREF voltage level is output on CVREF pin
0 = CVREF voltage level is disconnected from CVREF pin

bit 5 **CVRSS:** Comparator VREF Source Selection bit
1 = Comparator reference source, CVRSRC = VREF+ – VREF-
0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 4-0 **CVR<4:0>:** Comparator VREF Value Selection 0 ≤ CVR<4:0> ≤ 31 bits
When CVRSS = 1:
CVREF = (VREF-) + (CVR<4:0>/32) • (VREF+ – VREF-)
When CVRSS = 0:
CVREF = (AVSS) + (CVR<4:0>/32) • (AVDD – AVSS)

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)	$\overline{\text{PWRTEN}}$(2)
bit 7							bit 0

Legend:	P = Programmable 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 **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 a high-power level
01 = BORMV is set to a medium power level
00 = BORMV is set to a low-power level

bit 4-3 **BORV<1:0>:** Brown-out Reset Voltage bits(1)
11 = BVDD is set to 1.8V
10 = BVDD is set to 2.0V
01 = BVDD is set to 2.7V
00 = BVDD 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 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit(2)
1 = PWRT disabled
0 = PWRT enabled

Note 1: For the specifications, see **Section 31.1 “DC Characteristics: Supply Voltage PIC18F66K80 Family (Industrial/Extended)”**.

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

28.6.2 DATA EEPROM CODE PROTECTION

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

28.6.3 CONFIGURATION REGISTER PROTECTION

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

28.7 ID Locations

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

28.8 In-Circuit Serial Programming

The PIC18F66K80 family of devices can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

For the various programming modes, see the programming specification

28.9 In-Circuit Debugger

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

TABLE 28-5: DEBUGGER RESOURCES

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

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to $\overline{\text{MCLR}}$/RE3, V_{DD}, V_{SS}, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third-party development tool companies.

TSTFSZ	Test f, Skip if 0
Syntax:	TSTFSZ f {,a}
Operands:	$0 \le f \le 255$ $a \in [0,1]$
Operation:	skip if f = 0
Status Affected:	None
Encoding:	`0110` `011a` `ffff` `ffff`
Description:	If ‘f’ = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this a two-cycle instruction. If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank. If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.
Words:	1
Cycles:	1(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE     TSTFSZ  CNT, 1
NZERO    :
ZERO     :
```

Before Instruction
PC = Address (HERE)
After Instruction
If CNT = 00h,
PC = Address (ZERO)
If CNT ≠ 00h,
PC = Address (NZERO)

XORLW	Exclusive OR Literal with W
Syntax:	XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k → W
Status Affected:	N, Z
Encoding:	`0000` `1010` `kkkk` `kkkk`
Description:	The contents of W are XORed with the 8-bit literal ‘k’. The result is placed in W.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example:

```
XORLW   0AFh
```

Before Instruction
W = B5h
After Instruction
W = 1Ah

31.3 DC Characteristics: PIC18F66K80 Family (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**				
		All I/O Ports:				
D031		Schmitt Trigger Buffer	VSS	0.2 VDD	V	1.8V ≤ VDD ≤ 5.5V
D031A		RC3 and RC4	VSS	0.3 VDD	V	I2C™ enabled
D031B			VSS	0.8	V	SMBus enabled
D032		$\overline{\text{MCLR}}$	VSS	0.2 VDD	V	
D033		OSC1	VSS	0.2 VDD	V	LP, XT, HS modes
D033A		OSC1	VSS	0.2 VDD	V	EC modes
D034		SOSCI	VSS	0.3 VDD	V	
	VIH	**Input High Voltage**				
		All I/O Ports:				
D041		Schmitt Trigger Buffer	0.8	VDD	V	1.8V ≤ VDD ≤ 5.5V
D041A		RC3 and RC4	0.7 VDD	VDD	V	I2C enabled
D041B			2.1	VDD	V	SMBus enabled
D042		$\overline{\text{MCLR}}$	0.8 VDD	VDD	V	
D043		OSC1	0.9 VDD	VDD	V	RC mode
D043A		OSC1	0.7 VDD	VDD	V	HS mode
D044		SOSCI	0.7 VDD	VDD	V	
	IIL	**Input Leakage Current**(1)				
D060		I/O Ports	±50	±500	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
D061		$\overline{\text{MCLR}}$	—	±500	nA	Vss ≤ VPIN ≤ VDD
D063		OSC1	—	1	μA	Vss ≤ VPIN ≤ VDD
	IPU	**Weak Pull-up Current**				
D070		Weak Pull-up Current	50	400	μA	VDD = 5.5V, VPIN = VSS
	VOL	**Output Low Voltage**				
D080		I/O Ports:				
		PORTA, PORTB, PORTC	—	0.6	V	IOL = 8.5 mA, VDD = 5.5V, -40°C to +125°C
		PORTD, PORTE, PORTF, PORTG	—	0.6	V	IOL = 3.5 mA, VDD = 5.5V, -40°C to +125°C
D083		OSC2/CLKO (EC modes)	—	0.6	V	IOL = 1.6 mA, VDD = 5.5V, -40°C to +125°C

Note 1: Negative current is defined as current sourced by the pin.