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	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)	4V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°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-h-ss

TABLE 1-6: PIC18F6XK80 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Num	Pin Type	Buffer Type	Description
				PORTE is a bidirectional I/O port.
RE0/AN5/$\overline{RD}$	37			
RE0		I/O	ST/ CMOS	Digital I/O.
AN5		I	Analog	Analog Input 5.
$\overline{RD}$		I	ST	Parallel Slave Port read strobe.
RE1/AN6/C1OUT/$\overline{WR}$	38			
RE1		I/O	ST/ CMOS	Digital I/O.
AN6		I	Analog	Analog Input 6.
C1OUT		O	CMOS	Comparator 1 output.
$\overline{WR}$		I	ST	Parallel Slave Port write strobe.
RE2/AN7/C2OUT/$\overline{CS}$	39			
RE2		I/O	ST/ CMOS	Digital I/O.
AN7		I	Analog	Analog Input 7.
C2OUT		O	CMOS	Comparator 2 output.
$\overline{CS}$		I	ST	Parallel Slave Port chip select.
RE3				See the $\overline{MCLR}$/RE3 pin.
RE4/CANRX	27			
RE4		I/O	ST/ CMOS	Digital I/O.
CANRX		I	ST	CAN bus RX.
RE5/CANTX	24			
RE5		I/O	ST/ CMOS	Digital I/O.
CANTX		O	CMOS	CAN bus TX.
RE6/RX2/DT2	60			
RE6		I/O	ST/ CMOS	Digital I/O.
RX2		I	ST	EUSART asynchronous receive.
DT2		I/O	ST	EUSART synchronous data. (See related TX2/CK2.)
RE7/TX2/CK2	61			
RE7		I/O	ST/ CMOS	Digital I/O.
TX2		O	CMOS	EUSART asynchronous transmit.
CK2		I/O	ST	EUSART synchronous clock. (See related RX2/DT2.)

Legend: I^2C™ = I^2C/SMBus input buffer
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output

5.2 Master Clear Reset ($\overline{\text{MCLR}}$)

The $\overline{\text{MCLR}}$ pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the $\overline{\text{MCLR}}$ Reset path which detects and ignores small pulses.

The $\overline{\text{MCLR}}$ pin is not driven low by any internal Resets, including the WDT.

In PIC18F66K80 family devices, the $\overline{\text{MCLR}}$ input can be disabled with the MCLRE Configuration bit. When $\overline{\text{MCLR}}$ is disabled, the pin becomes a digital input. See **Section 11.6 "PORTE, TRISE and LATE Registers"** for more information.

5.3 Power-on Reset (POR)

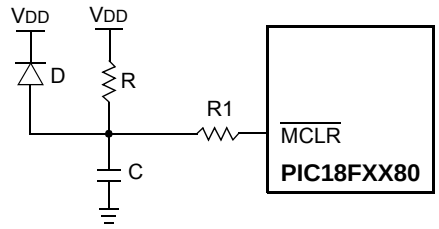
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the $\overline{\text{MCLR}}$ pin through a resistor (1 kΩ to 10 kΩ) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (Parameter D004). For a slow rise time, see Figure 5-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the $\overline{\text{POR}}$ bit (RCON<1>). The state of the bit is set to '0' whenever a Power-on Reset occurs; it does not change for any other Reset event. $\overline{\text{POR}}$ is not reset to '1' by any hardware event. To capture multiple events, the user manually resets the bit to '1' in software following any Power-on Reset.

FIGURE 5-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode, D, helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 ≥ 1 kΩ will limit any current flowing into $\overline{\text{MCLR}}$ from external capacitor, C, in the event of $\overline{\text{MCLR}}$/VPP pin breakdown, due to Electro-static Discharge (ESD) or Electrical Overstress (EOS).

6.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit Program Counter (PC) that is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The entire PIC18F66K80 family offers a range of on-chip Flash program memory sizes, from 32 Kbytes (16,384 single-word instructions) to 64 Kbytes (32,768 single-word instructions).

- PIC18F25K80, PIC18F45K80 and PIC18F65K80 – 32 Kbytes of Flash memory, storing up to 16,384 single-word instructions
- PIC18F26K80, PIC18F46K80 and PIC18F66K80 – 64 Kbytes of Flash memory, storing up to 32,768 single-word instructions

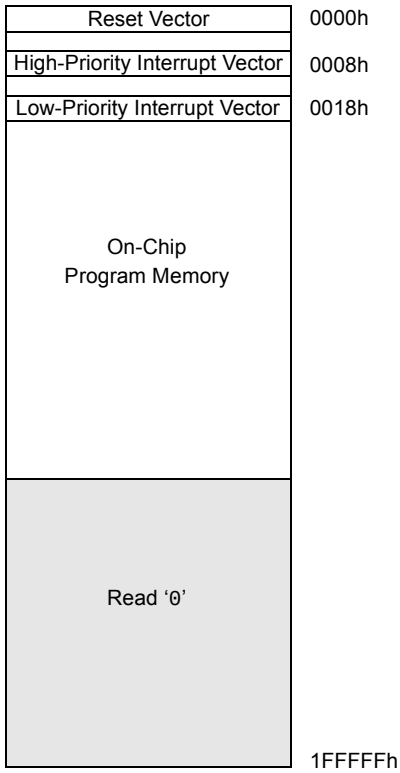
The program memory maps for individual family members are shown in Figure 6-1.

6.1.1 HARD MEMORY VECTORS

All PIC18 devices have a total of three hard-coded return vectors in their program memory space. The Reset vector address is the default value to which the Program Counter returns on all device Resets. It is located at 0000h.

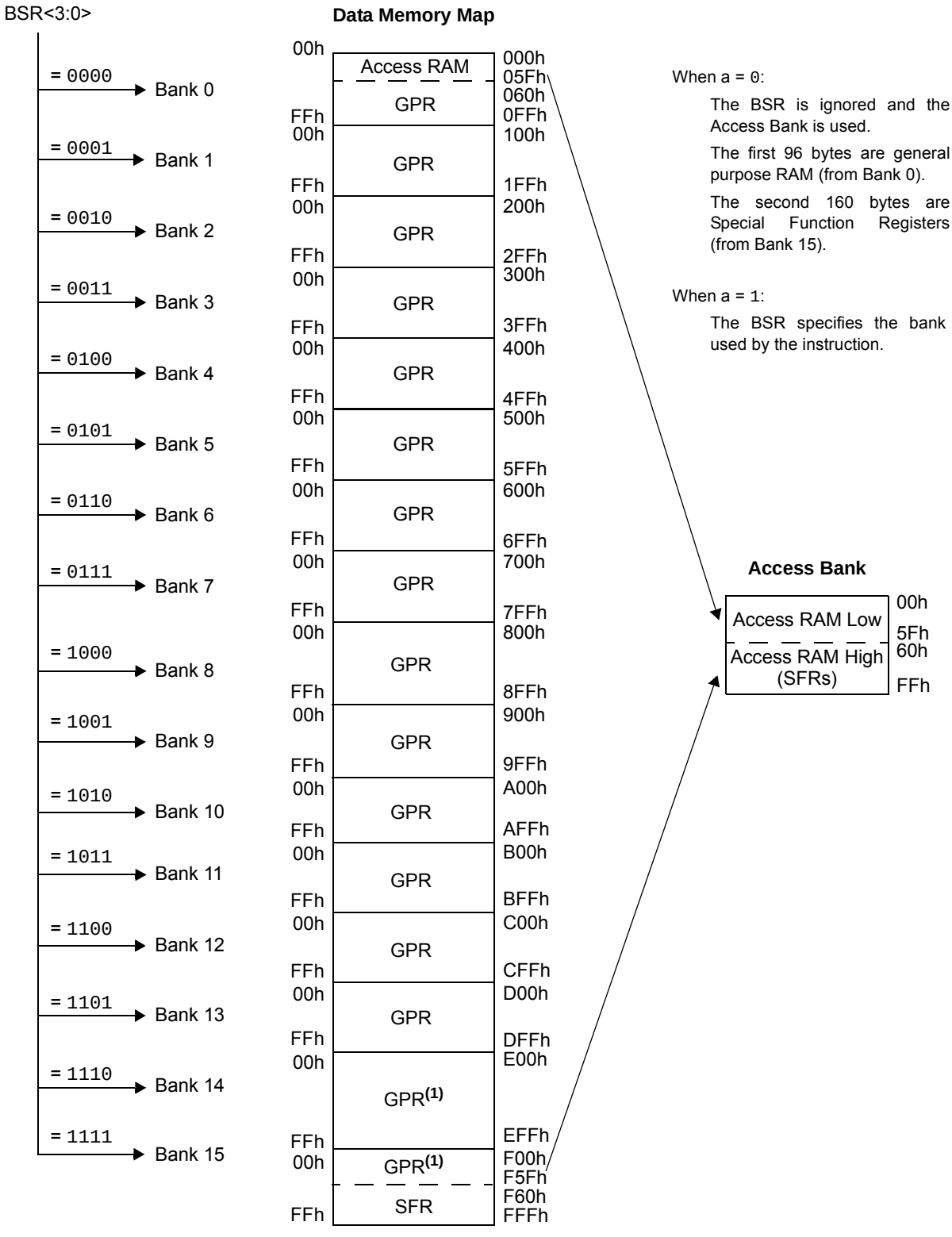
PIC18 devices also have two interrupt vector addresses for handling high-priority and low-priority interrupts. The high-priority interrupt vector is located at 0008h and the low-priority interrupt vector is at 0018h. The locations of these vectors are shown, in relation to the program memory map, in Figure 6-2.

FIGURE 6-2: HARD VECTOR FOR PIC18F66K80 FAMILY DEVICES



Legend: (Top of Memory) represents upper boundary of on-chip program memory space (see Figure 6-1 for device-specific values). Shaded area represents unimplemented memory. Areas are not shown to scale.

FIGURE 6-6: DATA MEMORY MAP FOR PIC18FX5K80 AND PIC18FX6K80 DEVICES



Note 1: Addresses, E41h through F5Fh, are also used by SFRs, but are not part of the Access RAM. Users must always use the complete address, or load the proper BSR value, to access these registers.

TABLE 11-9: PORTE FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE5/CANTX	RE5(1)	0	O	DIG	LATE<5> data output.
		1	I	ST	PORTE<5> data input.
	CANTX(1,2)	0	O	DIG	CAN bus TX.
RE6/RX2/DT2	RE6(1)	0	O	DIG	LATE<6> data output.
		1	I	ST	PORTE<6> 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.
RE7/TX2/CK2	RE7(1)	0	O	DIG	LATE<7> data output.
		1	I	ST	PORTE<7> data input.
	TX2(1)	0	O	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.
	CK2(1)	0	O	DIG	Synchronous serial clock output (EUSARTx module); user must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSARTx module); user must configure as an input.

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: These bits are unavailable for 40 and 44-pin devices (PIC18F4XK0).

2: This is the alternate pin assignment for CANRX and CANTX on 64-pin devices (PIC18F6XK80) when the CANMX Configuration bit is cleared.

TABLE 11-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTE	RE7(1)	RE6(1)	RE5(1)	RE4(1)	RE3	RE2	RE1	RE0
LATE	LATE7	LATE6	LATE5	LATE4	—	LATE2	LATE1	LATE0
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	—	TRISE2	TRISE1	TRISE0
PADCFG1	RDPU	REPU	RFPU(1)	RGPU(1)	—	—	—	CTMUDS
ANCON0	ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0

Legend: Shaded cells are not used by PORTE.

Note 1: These bits are unimplemented on 44-pin devices, read as '0'.

15.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- Eight-bit Timer and Period registers (TMR2 and PR2, respectively)
- Both registers are readable and writable
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register (Register 15-1) that enables or disables the timer, and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 15-1.

15.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A four-bit counter/prescaler on the clock input gives the prescale options of direct input, divide-by-4 or divide-by-16. These are selected by the prescaler control bits, T2CKPS<1:0> (T2CON<1:0>).

The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler. (See **Section 15.2 "Timer2 Interrupt"**.)

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset – Power-on Reset (POR), MCLR Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR)

TMR2 is not cleared when T2CON is written.

Note:	The CCP and ECCP modules use Timers, 1 through 4, 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 CCPTMRS register. For more details, see Register 20-2 and Register 19-2.

REGISTER 15-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

18.8 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 18-5 for an example circuit. When CTMUDS (PADCFG1<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 18.4.1 "Current Source Calibration"**) and V is the internal reference voltage (CVREF).

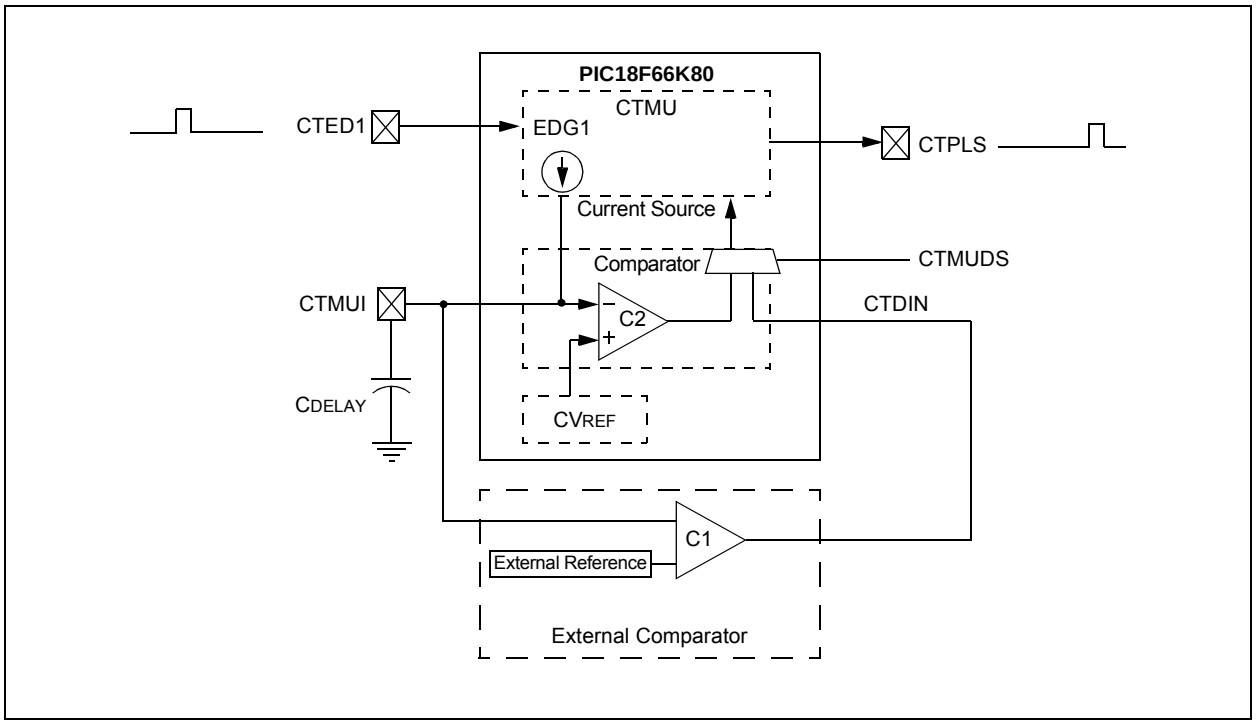
An example use of the external capacitor feature is interfacing with variable capacitive-based sensors, such as a humidity sensor. As the humidity varies, the pulse-width output on CTPLS will vary. An example use of the CTDIN feature is interfacing with a digital sensor. The CTPLS output pin can be connected 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 18-5: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



19.4 PWM Mode

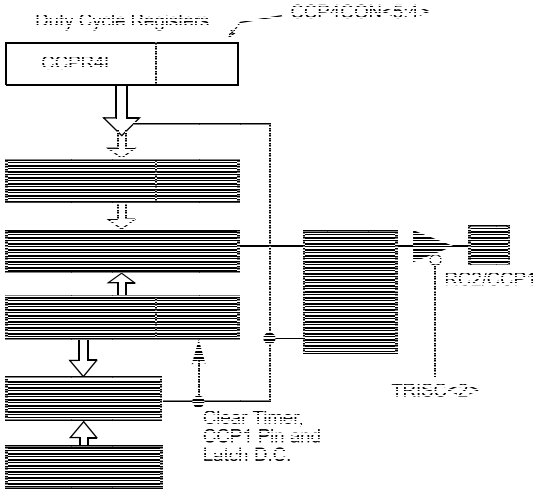
In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCPx pin is multiplexed with a PORTC or PORTB data latch, the appropriate TRIS bit must be cleared to make the CCPx pin an output.

Note:	Clearing the CCPxCON register will force the corresponding CCPx output latch (depending on device configuration) to the default low level. This is not the PORTx I/O data latch.

Figure 19-3 shows a simplified block diagram of the CCPx module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 19.4.3 "Setup for PWM Operation"**.

FIGURE 19-3: SIMPLIFIED PWM BLOCK DIAGRAM

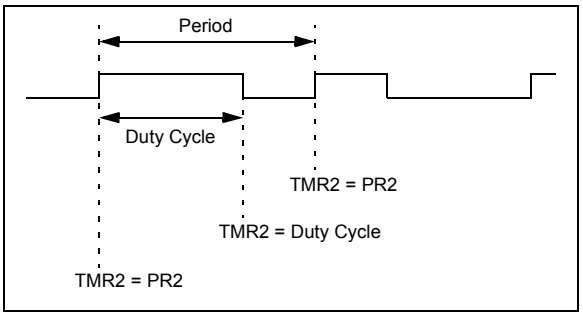


Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

2: CCP4 and its appropriate timers are used as an example. For details on all of the CCP modules and their timer assignments, see Table 19-2.

A PWM output (Figure 19-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 19-4: PWM OUTPUT



19.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

EQUATION 19-1:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP4 pin is set

  (An exception: If PWM duty cycle = 0%, the CCP4 pin will not be set)
- The PWM duty cycle is latched from CCPR4L into CCPR4H

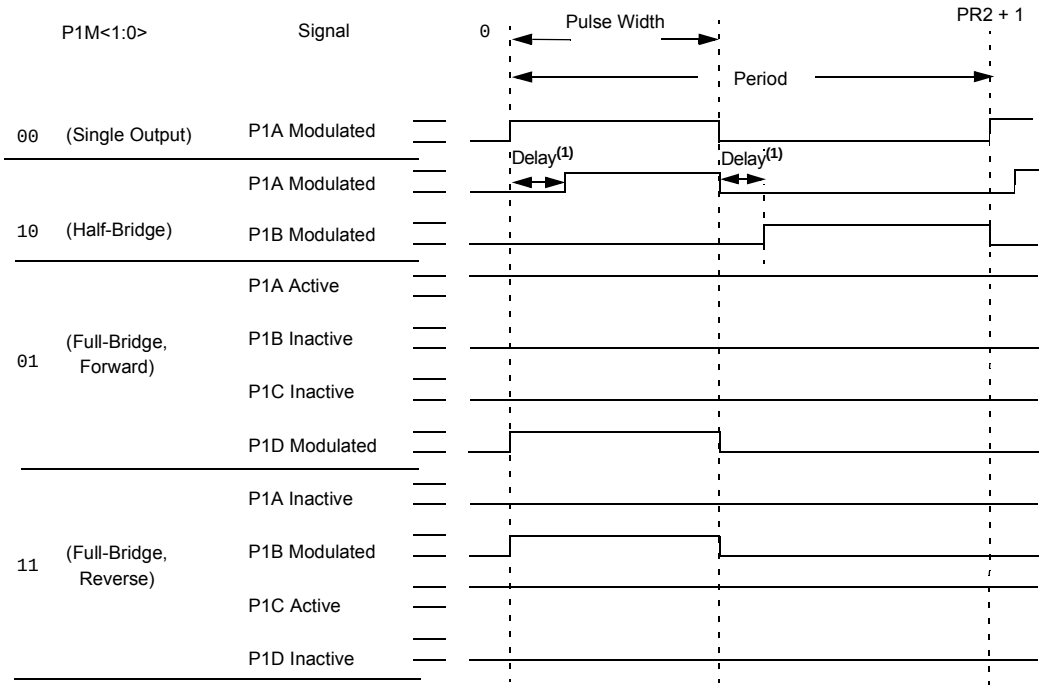
Note:	The Timer2 postscalers (see **Section 15.0 "Timer2 Module"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

TABLE 20-2: EXAMPLE PIN ASSIGNMENTS FOR VARIOUS PWM ENHANCED MODES

ECCP Mode	P1M<1:0>	P1A	P1B	P1C	P1D
Single	00	Yes(1)	Yes(1)	Yes(1)	Yes(1)
Half-Bridge	10	Yes	Yes	No	No
Full-Bridge, Forward	01	Yes	Yes	Yes	Yes
Full-Bridge, Reverse	11	Yes	Yes	Yes	Yes

Note 1: Outputs are enabled by pulse steering in Single mode (see Register 20-5).

FIGURE 20-4: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)



Relationships:
- Period = 4 * TOSC * (PR2 + 1) * (TMR2 Prescale Value)
- Pulse Width = TOSC * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * TOSC * (ECCP1DEL<6:0>)

Note 1: Dead-band delay is programmed using the ECCP1DEL register (**Section 20.4.6 "Programmable Dead-Band Delay Mode"**).

21.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP module consists of a Transmit/Receive Shift register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 21-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the SSPSTAT register indicates the various status conditions.

21.3.3 OPEN-DRAIN OUTPUT OPTION

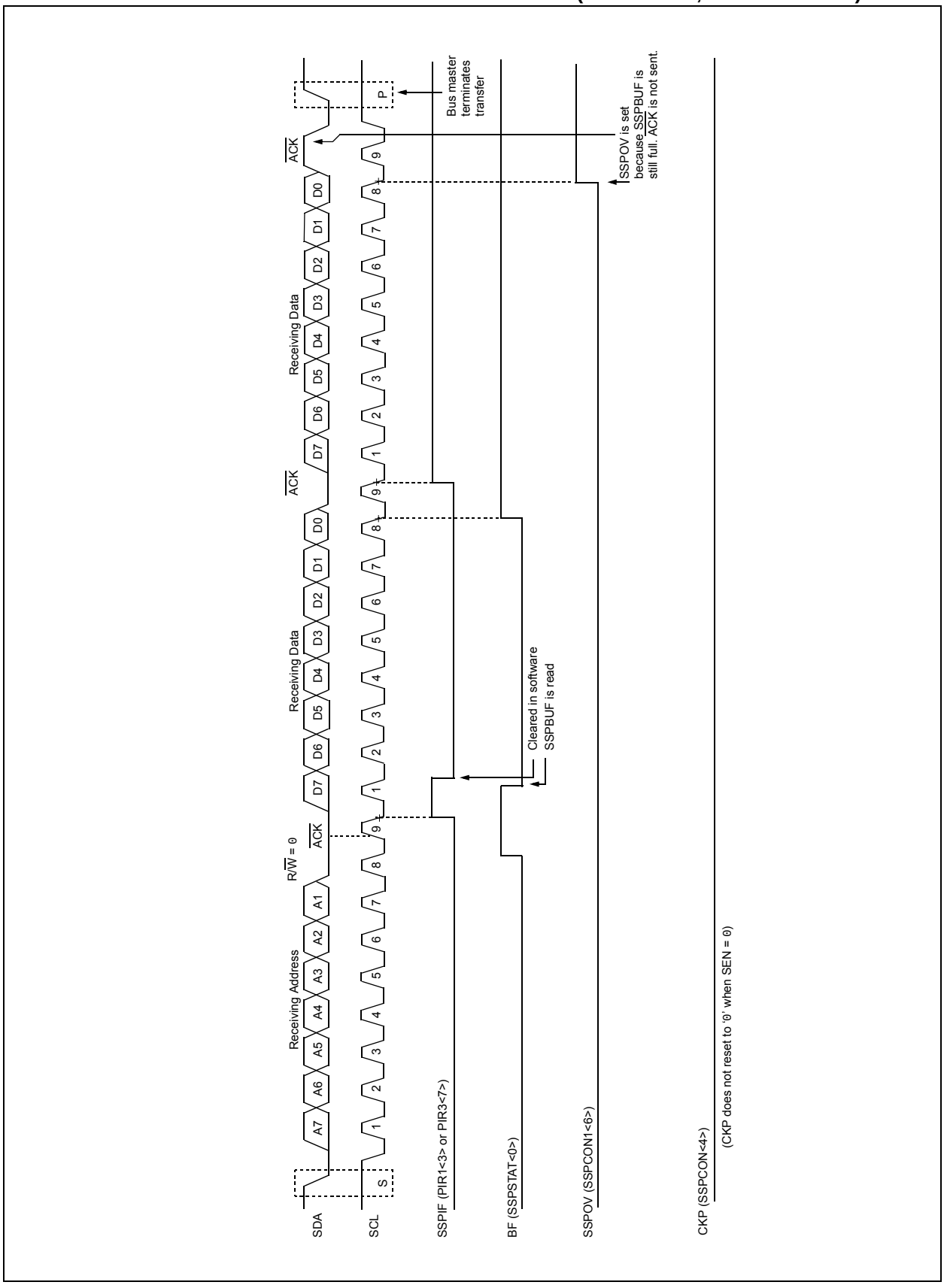
The drivers for the SDO output and SCK clock pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters. For more information, see **Section 11.1.3 "Open-Drain Outputs"**.

The open-drain output option is controlled by the SSPOD bit (ODCON<7>). Setting the SSPOD bit configures the SDO and SCK pins for open-drain operation.

EXAMPLE 21-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP    BTFSS   SSPSTAT, BF     ;Has data been received (transmit complete)?
        BRA     LOOP            ;No
        MOVF    SSPBUF, W       ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSPBUF          ;New data to xmit
```

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



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 SDA while SCL 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 SDA, while SCL 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 SCL 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 (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

21.4.14 SLEEP OPERATION

While in Sleep mode, the I^2C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

21.4.15 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

21.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

21.4.17 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high, and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF, and reset the I^2C port to its Idle state (Figure 21-27).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

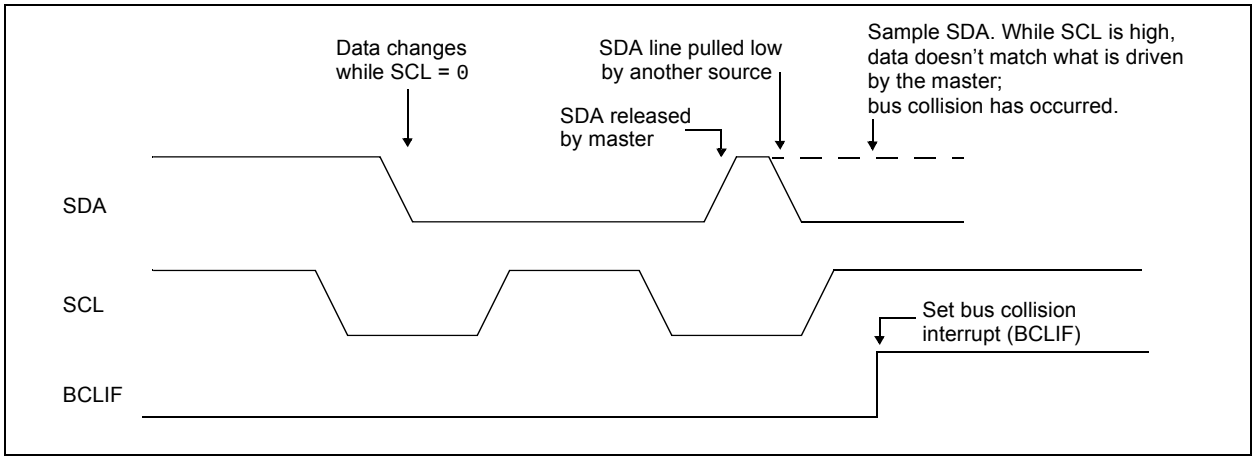
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine, and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 21-27: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



REGISTER 22-2: RCSTAx: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit

1 = Serial port is enabled (configures RXx/DTx and TXx/CKx pins as serial port pins)
0 = Serial port is disabled (held in Reset)

bit 6 **RX9:** 9-Bit Receive Enable bit

1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit

Asynchronous mode:
Don't care.

Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.

Synchronous mode – Slave:
Don't care.

bit 4 **CREN:** Continuous Receive Enable bit

Asynchronous mode:
1 = Enables receiver
0 = Disables receiver

Synchronous mode:
1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit

Asynchronous mode 9-Bit (RX9 = 1):
1 = Enables address detection; enables interrupt and loads the receive buffer when RSR<8> is set
0 = Disables address detection; all bytes are received and the ninth bit can be used as a parity bit

Asynchronous mode 9-Bit (RX9 = 0):
Don't care.

bit 2 **FERR:** Framing Error bit

1 = Framing error (can be cleared by reading the RCREGx register and receiving next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit

1 = Overrun error (can be cleared by clearing bit, CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data

This can be address/data bit or a parity bit and must be calculated by user firmware.

REGISTER 23-7: ADRESL: A/D RESULT LOW BYTE REGISTER, RIGHT JUSTIFIED (ADFM = 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
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit 7							

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

bit 7-0 **ADRES<7:0>:** A/D Result Low Byte bits

The ANCONx registers are used to configure the operation of the I/O pin associated with each analog channel. Clearing an ANSELx bit configures the corresponding pin (ANx) to operate as a digital only I/O. Setting a bit configures the pin to operate as an analog input for either the A/D Converter or the comparator module, with all digital peripherals disabled and digital inputs read as '0'.

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

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

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSEL7(1)	ANSEL6(1)	ANSEL5(1)	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
bit 7							bit 0

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

bit 7-0 **ANSEL<7:0>:** Analog Port Configuration bits (AN7 and AN0)(1)

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

Note 1: AN14 through AN11 and AN7 to AN5 are implemented only on 40/44-pin and 64-pin devices. For 28-pin devices, the corresponding ANSELx bits are still implemented for these channels, but have no effect.

REGISTER 27-23: BnCON: TX/RX BUFFER 'n' CONTROL REGISTERS IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 1](1)

R/W-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TXBIF(3)	TXABT(3)	TXLARB(3)	TXERR(3)	TXREQ(2,4)	RTREN	TXPRI1(5)	TXPRI0(5)
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 **TXBIF:** Transmit Buffer Interrupt Flag bit(3)
- 1 = A message was successfully transmitted
- 0 = No message was transmitted

bit 6 **TXABT:** Transmission Aborted Status bit(3)
- 1 = Message was aborted
- 0 = Message was not aborted

bit 5 **TXLARB:** Transmission Lost Arbitration Status bit(3)
- 1 = Message lost arbitration while being sent
- 0 = Message did not lose arbitration while being sent

bit 4 **TXERR:** Transmission Error Detected Status bit(3)
- 1 = A bus error occurred while the message was being sent
- 0 = A bus error did not occur while the message was being sent

bit 3 **TXREQ:** Transmit Request Status bit(2,4)
- 1 = Requests sending a message; clears the TXABT, TXLARB and TXERR bits
- 0 = Automatically cleared when the message is successfully sent

bit 2 **RTREN:** Automatic Remote Transmission Request Enable bit
- 1 = When a remote transmission request is received, TXREQ will be automatically set
- 0 = When a remote transmission request is received, TXREQ will be unaffected

bit 1-0 **TXPRI<1:0>:** Transmit Priority bits(5)
- 11 = Priority Level 3 (highest priority)
- 10 = Priority Level 2
- 01 = Priority Level 1
- 00 = Priority Level 0 (lowest priority)

Note 1: These registers are available in Mode 1 and 2 only.
2: Clearing this bit in software while the bit is set will request a message abort.
3: This bit is automatically cleared when TXREQ is set.
4: While TXREQ is set or a transmission is in progress, Transmit Buffer registers remain read-only.
5: These bits set the order in which the Transmit Buffer register will be transferred. They do not alter the CAN message identifier.

27.9 Baud Rate Setting

All nodes on a given CAN bus must have the same nominal bit rate. The CAN protocol uses Non-Return-to-Zero (NRZ) coding which does not encode a clock within the data stream. Therefore, the receive clock must be recovered by the receiving nodes and synchronized to the transmitter's clock.

As oscillators and transmission time may vary from node to node, the receiver must have some type of Phase Lock Loop (PLL) synchronized to data transmission edges to synchronize and maintain the receiver clock. Since the data is NRZ coded, it is necessary to include bit stuffing to ensure that an edge occurs at least every six bit times to maintain the Digital Phase Lock Loop (DPLL) synchronization.

The bit timing of the PIC18F66K80 family is implemented using a DPLL that is configured to synchronize to the incoming data and provides the nominal timing for the transmitted data. The DPLL breaks each bit time into multiple segments made up of minimal periods of time called the *Time Quanta* (TQ).

Bus timing functions executed within the bit time frame, such as synchronization to the local oscillator, network transmission delay compensation and sample point positioning, are defined by the programmable bit timing logic of the DPLL.

All devices on the CAN bus must use the same bit rate. However, all devices are not required to have the same master oscillator clock frequency. For the different clock frequencies of the individual devices, the bit rate has to be adjusted by appropriately setting the baud rate prescaler and number of time quanta in each segment.

The "Nominal Bit Rate" is the number of bits transmitted per second, assuming an ideal transmitter with an ideal oscillator, in the absence of resynchronization. The nominal bit rate is defined to be a maximum of 1 Mb/s.

The "Nominal Bit Time" is defined as:

EQUATION 27-1: NOMINAL BIT TIME

$$T_{BIT} = 1/\text{Nominal Bit Rate}$$

The Nominal Bit Time can be thought of as being divided into separate, non-overlapping time segments. These segments (Figure 27-4) include:

- Synchronization Segment (Sync_Seg)
- Propagation Time Segment (Prop_Seg)
- Phase Buffer Segment 1 (Phase_Seg1)
- Phase Buffer Segment 2 (Phase_Seg2)

The time segments (and thus, the Nominal Bit Time) are, in turn, made up of integer units of time called Time Quanta or TQ (see Figure 27-4). By definition, the Nominal Bit Time is programmable from a minimum of 8 TQ to a maximum of 25 TQ. Also by definition, the minimum Nominal Bit Time is 1 μs, corresponding to a maximum 1 Mb/s rate. The actual duration is given by the following relationship:

EQUATION 27-2: NOMINAL BIT TIME DURATION

$$\text{Nominal Bit Time} = T_Q * (\text{Sync\_Seg} + \text{Prop\_Seg} + \text{Phase\_Seg1} + \text{Phase\_Seg2})$$

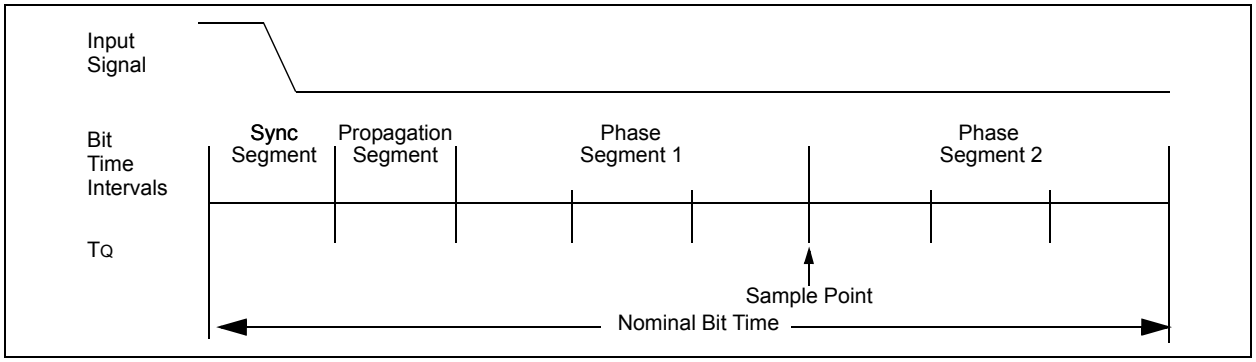
The Time Quantum is a fixed unit derived from the oscillator period. It is also defined by the programmable baud rate prescaler, with integer values from 1 to 64, in addition to a fixed divide-by-two for clock generation. Mathematically, this is:

EQUATION 27-3: TIME QUANTUM

$$T_Q\ (\mu s) = (2 * (BRP + 1))/F_{OSC}\ (MHz)$$
or
$$T_Q\ (\mu s) = (2 * (BRP + 1)) * T_{OSC}\ (\mu s)$$

where FOSC is the clock frequency, TOSC is the corresponding oscillator period and BRP is an integer (0 through 63) represented by the binary values of BRGCON1<5:0>. The equation above refers to the effective clock frequency used by the microcontroller. If, for example, a 10 MHz crystal in HS mode is used, then FOSC = 10 MHz and TOSC = 100 ns. If the same 10 MHz crystal is used in HS-PLL mode, then the effective frequency is FOSC = 40 MHz and TOSC = 25 ns.

FIGURE 27-4: BIT TIME PARTITIONING



REGISTER 28-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

U-0	R/P-1	U-0	R/P-1	R/P-1	R/P-1	U-0	R/P-1
—	XINST	—	SOSCSEL1	SOSCSEL0	INTOSCSEL	—	$\overline{\text{RETEN}}$
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 **XINST:** Extended Instruction Set Enable bit

1 = Instruction set extension and Indexed Addressing mode are enabled
0 = Instruction set extension and Indexed Addressing mode are disabled (Legacy mode)

bit 5 **Unimplemented:** Read as ‘0’

bit 4-3 **SOSCSEL<1:0>:** SOSC Power Selection and Mode Configuration bits

11 = High-power SOSC circuit is selected
10 = Digital (SCLKI) mode; I/O port functionality of RC0 and RC1 is enabled
01 = Low-power SOSC circuit is selected
00 = Reserved

bit 2 **INTOSCSEL:** LF-INTOSC Low-power Enable bit

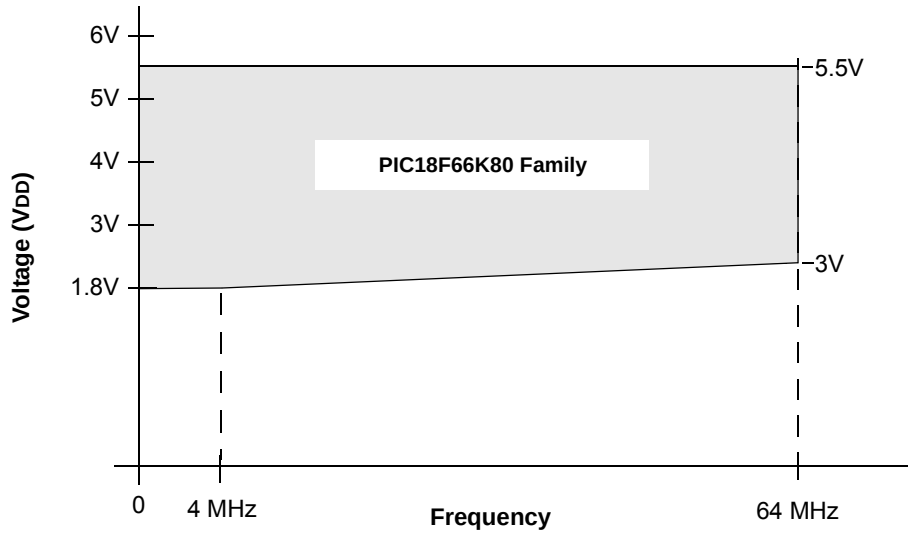
1 = LF-INTOSC in High-Power mode during Sleep
0 = LF-INTOSC in Low-Power mode during Sleep

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **$\overline{\text{RETEN}}$:** VREG Sleep Enable bit

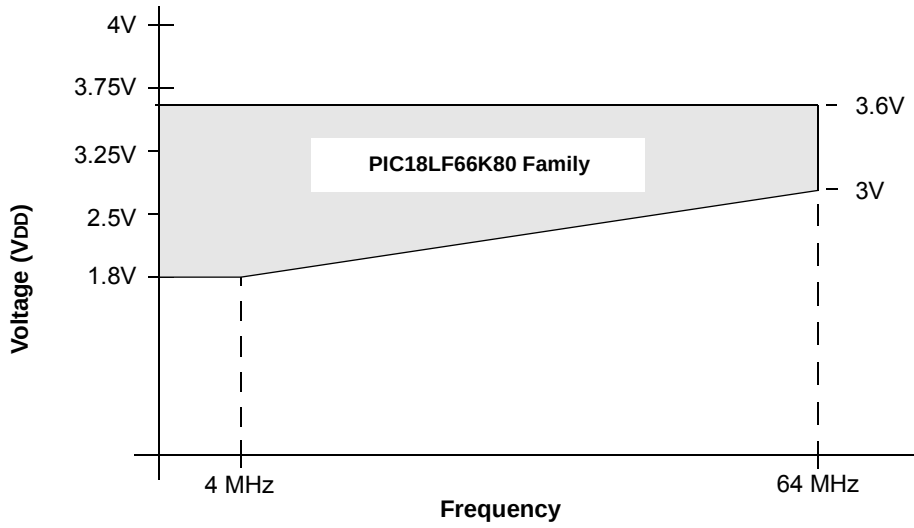
1 = Ultra low-power regulator is disabled. Regulator power in Sleep mode is controlled by REGSLP (WDTCON<7>).
0 = Ultra low-power regulator is enabled. Regulator power in Sleep mode is controlled by SRETEN (WDTCON<4>).

FIGURE 31-1: VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL/EXTENDED)(1)



Note 1: For VDD values 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

FIGURE 31-2: VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL/EXTENDED)(1,2)



Note 1: When the on-chip voltage regulator is disabled, VDD must be maintained so that VDD ≤ 3.6V.
2: For VDD values 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

TABLE 31-1: MEMORY PROGRAMMING REQUIREMENTS

DC CHARACTERISTICS			Standard Operating Conditions Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Internal Program Memory Programming Specifications(1)					
D110	VPP	Voltage on $\overline{\text{MCLR}}$/VPP/RE5 pin	VDD + 1.5	—	10	V	**(Note 3, Note 4)**
D113	IDDP	Supply Current during Programming	—	—	10	mA	
		Data EEPROM Memory					**(Note 2)**
D120	ED	Byte Endurance	100K	1000K	—	E/W	-40°C to +125°C
D121	VDRW	VDD for Read/Write	1.8	—	5.5	V	Using EECON to read/write PIC18FXXKXX devices
			1.8	—	3.6	V	Using EECON to read/write PIC18LFXXKXX devices
D122	TDEW	Erase/Write Cycle Time	—	4	—	ms	
D123	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh(2)	1M	10M	—	E/W	-40°C to +125°C
		Program Flash Memory					
D130	EP	Cell Endurance	1K	10K	—	E/W	-40°C to +125°C
D131	VPR	VDD for Read	1.8	—	5.5	V	PIC18FXXKXX devices
			1.8	—	3.6	V	PIC18LFXXKXX devices
D132B	VPEW	Voltage for Self-Timed Erase or Write Operations VDD	1.8	—	5.5	V	PIC18FXXKXX devices
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	—	10	mA	
D140	TWE	Writes per Erase Cycle	—	—	1		For each physical address

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: These specifications are for programming the on-chip program memory through the use of table write instructions.

2: Refer to **Section 8.8 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

3: Required only if Single-Supply Programming is disabled.

4: The MPLAB® ICD 2 does not support variable VPP output. Circuitry to limit the ICD2 VPP voltage must be placed between the ICD 2 and target system when programming or debugging with the ICD2.