



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	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f448-e-pt

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
RD0/PSP0/C1IN+ RD0 PSP0 C1IN+	—	19	38	21	I/O I/O I	ST TTL Analog	PORTD is a bidirectional I/O port. These pins have TTL input buffers when external memory is enabled. Digital I/O. Parallel Slave Port data. Comparator 1 input.
RD1/PSP1/C1IN- RD1 PSP1 C1IN-	—	20	39	22	I/O I/O I	ST TTL Analog	Digital I/O. Parallel Slave Port data. Comparator 1 input.
RD2/PSP2/C2IN+ RD2 PSP2 C2IN+	—	21	40	23	I/O I/O I	ST TTL Analog	Digital I/O. Parallel Slave Port data. Comparator 2 input.
RD3/PSP3/C2IN- RD3 PSP3 C2IN-	—	22	41	24	I/O I/O I	ST TTL Analog	Digital I/O. Parallel Slave Port data. Comparator 2 input.
RD4/PSP4/ECCP1/P1A RD4 PSP4 ECCP1 P1A	—	27	2	30	I/O I/O I/O O	ST TTL ST —	Digital I/O. Parallel Slave Port data. ECCP1 capture/compare. ECCP1 PWM output A.
RD5/PSP5/P1B RD5 PSP5 P1B	—	28	3	31	I/O I/O O	ST TTL —	Digital I/O. Parallel Slave Port data. ECCP1 PWM output B.
RD6/PSP6/P1C RD6 PSP6 P1C	—	29	4	32	I/O I/O O	ST TTL —	Digital I/O. Parallel Slave Port data. ECCP1 PWM output C.
RD7/PSP7/P1D RD7 PSP7 P1D	—	30	5	33	I/O I/O O	ST TTL —	Digital I/O. Parallel Slave Port data. ECCP1 PWM output D.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

PIC18FXX8

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Reset WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
TOSU	PIC18F2X8	PIC18F4X8	---0 0000	---0 0000	---0 uuuu ⁽³⁾
TOSH	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
TOSL	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
STKPTR	PIC18F2X8	PIC18F4X8	00-0 0000	uu-0 0000	uu-u uuuu ⁽³⁾
PCLATU	PIC18F2X8	PIC18F4X8	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	PC + 2 ⁽²⁾
TBLPTRU	PIC18F2X8	PIC18F4X8	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F2X8	PIC18F4X8	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
INTCON2	PIC18F2X8	PIC18F4X8	111- -1-1	111- -1-1	uuu- -u-u ⁽¹⁾
INTCON3	PIC18F2X8	PIC18F4X8	11-0 0-00	11-0 0-00	uu-u u-uu ⁽¹⁾
INDF0	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
POSTINC0	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
POSTDEC0	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
PREINC0	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
PLUSW0	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
FSR0H	PIC18F2X8	PIC18F4X8	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
POSTINC1	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
POSTDEC1	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
PREINC1	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A
PLUSW1	PIC18F2X8	PIC18F4X8	N/A	N/A	N/A

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 3-2 for Reset value for specific condition.
- 5: Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.
- 6: Values for CANSTAT also apply to its other instances (CANSTATRO1 through CANSTATRO4).

PIC18FXX8

5.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

Generally, a write failure will be a bit which was written as a '1', but reads back as a '0' (due to leakage off the cell).

5.6 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together reduce the probability of an accidental write during brown-out, power glitch or software malfunction.

5.7 Operation During Code-Protect

Data EEPROM memory has its own code-protect mechanism. External read and write operations are disabled if either of these mechanisms are enabled.

The microcontroller itself can both read and write to the internal data EEPROM, regardless of the state of the code-protect configuration bit. Refer to **Section 24.0 "Special Features of the CPU"** for additional information.

5.8 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). Frequently changing values will typically be updated more often than specification D124 or D124A. If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory. A simple data EEPROM refresh routine is shown in Example 5-3.

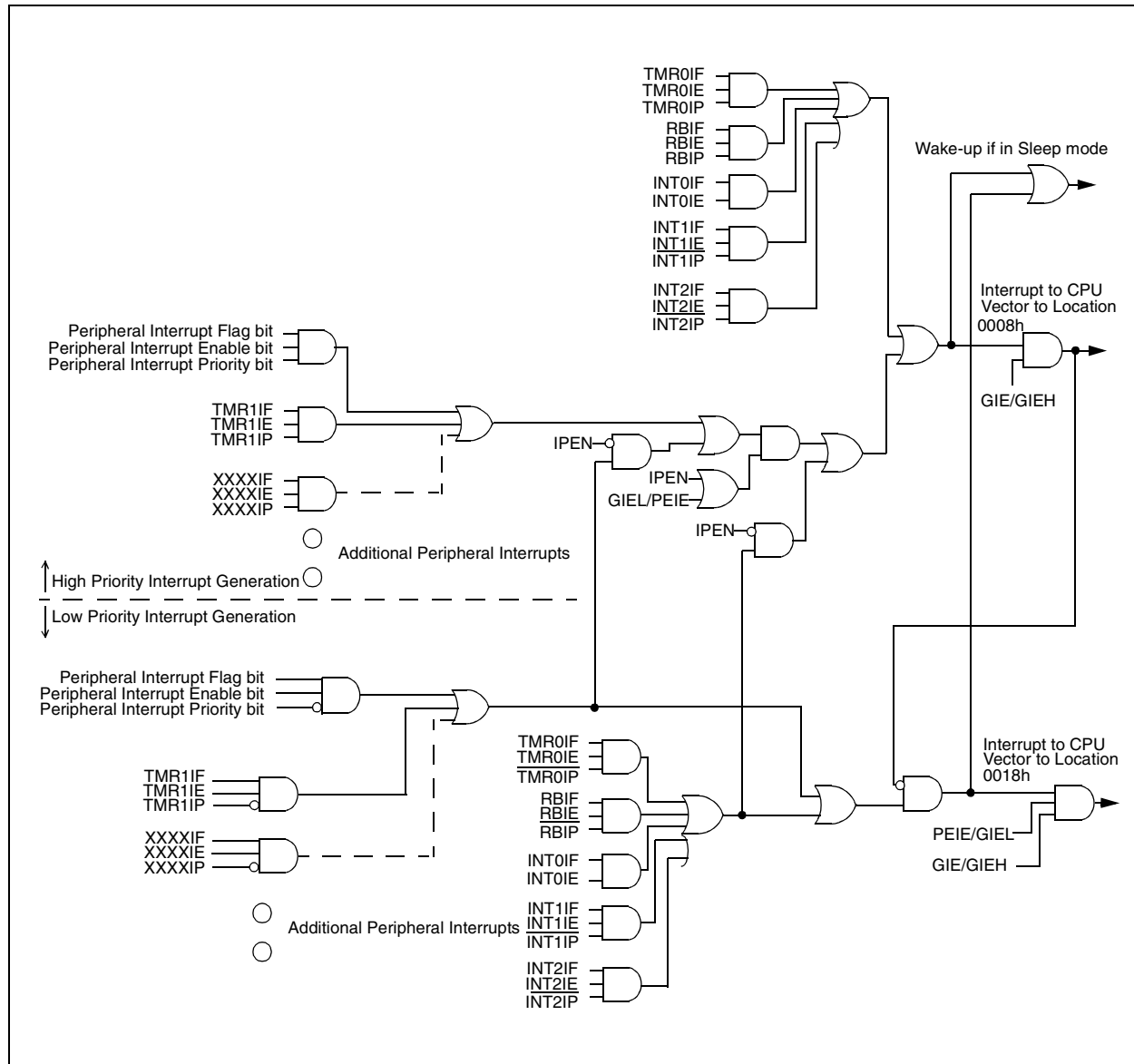
Note: If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification D124 or D124A.

EXAMPLE 5-3: DATA EEPROM REFRESH ROUTINE

```
CLRF    EEADR      ; Start at address 0
BCF     EECON1, CFGS ; Set for memory
BCF     EECON1, EEPGD ; Set for Data EEPROM
BCF     INTCON, GIE  ; Disable interrupts
BSF     EECON1, WREN  ; Enable writes
Loop    ; Loop to refresh array
BSF     EECON1, RD     ; Read current address
MOVLW   55h           ;
MOVWF   EECON2         ; Write 55h
MOVLW   0AAh          ;
MOVWF   EECON2         ; Write AAh
BSF     EECON1, WR     ; Set WR bit to begin write
BTFSC   EECON1, WR     ; Wait for write to complete
BRA     $-2
        ;
INCF    EEADR, F       ; Increment address
BRA     Loop           ; Not zero, do it again
        ;
BCF     EECON1, WREN   ; Disable writes
BSF     INTCON, GIE    ; Enable interrupts
```

PIC18FXX8

FIGURE 8-1: INTERRUPT LOGIC



PIC18FXX8

15.4 PWM Mode

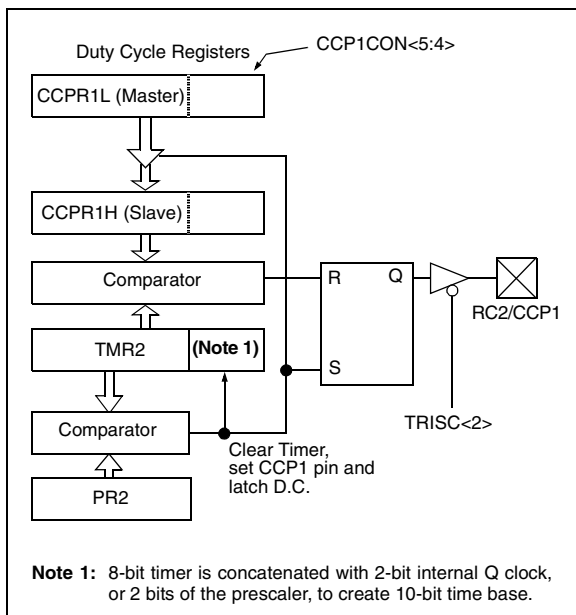
In Pulse-Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

Note: Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 15-3 shows a simplified block diagram of the CCP module in PWM mode.

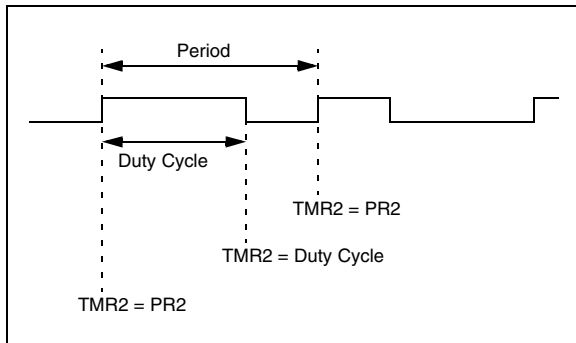
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.3 “Setup for PWM Operation”**.

FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 15-4) 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 15-4: PWM OUTPUT



15.4.1 PWM PERIOD

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

EQUATION 15-1:

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

PWM frequency is defined as $1/[\text{PWM period}]$.

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

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

Note: The Timer2 postscaler (see **Section 13.0 “Timer2 Module”**) is 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.

15.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSBs and the CCP1CON<5:4> contains the two LSBs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time.

EQUATION 15-2:

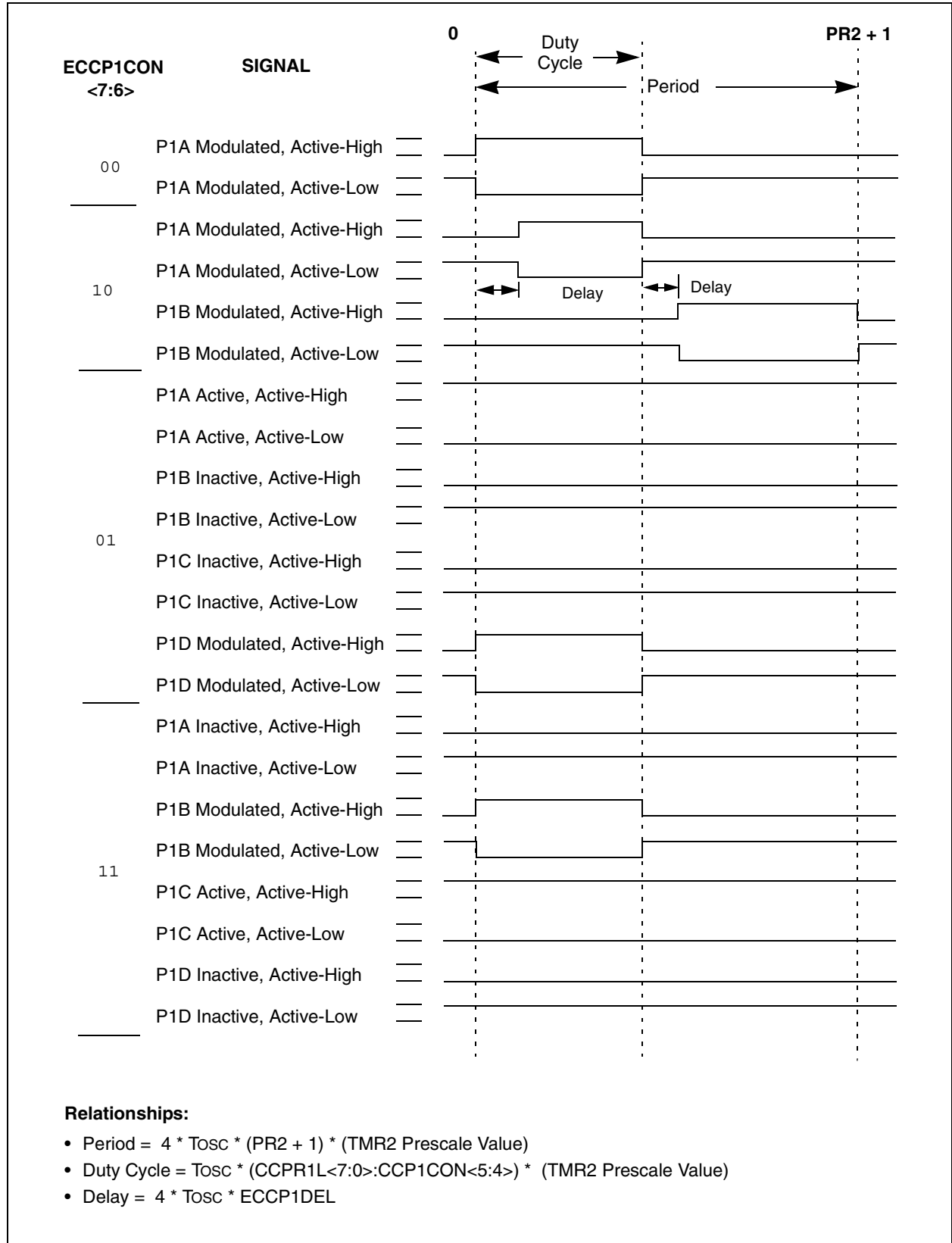
$$\text{PWM Duty Cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

FIGURE 16-2: PWM OUTPUT RELATIONSHIPS



PIC18FXX8

17.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 17-23).

17.4.12.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

17.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 17-24).

17.4.13.1 WCOL Status Flag

If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 17-23: ACKNOWLEDGE SEQUENCE WAVEFORM

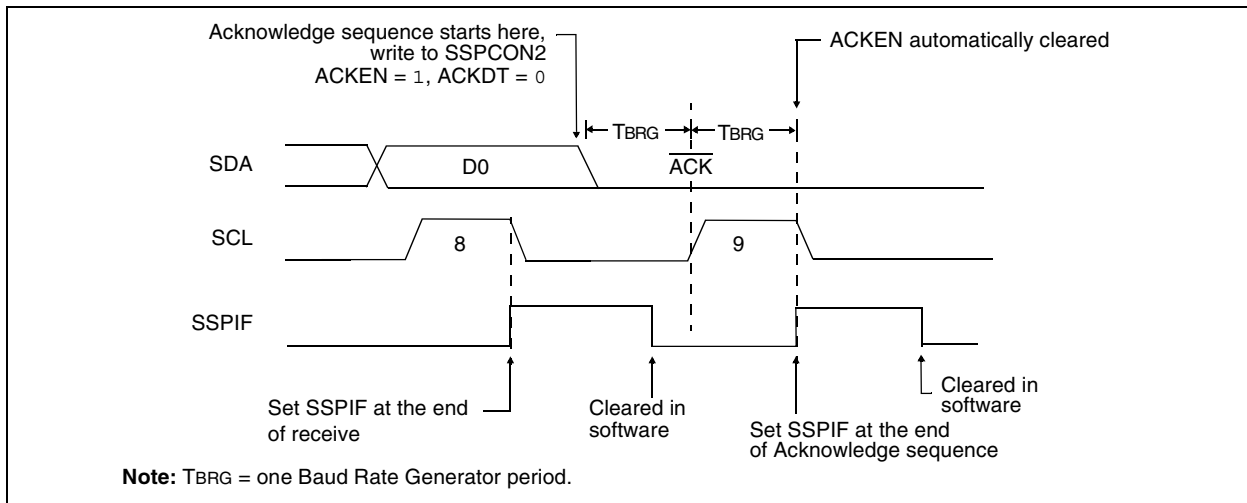
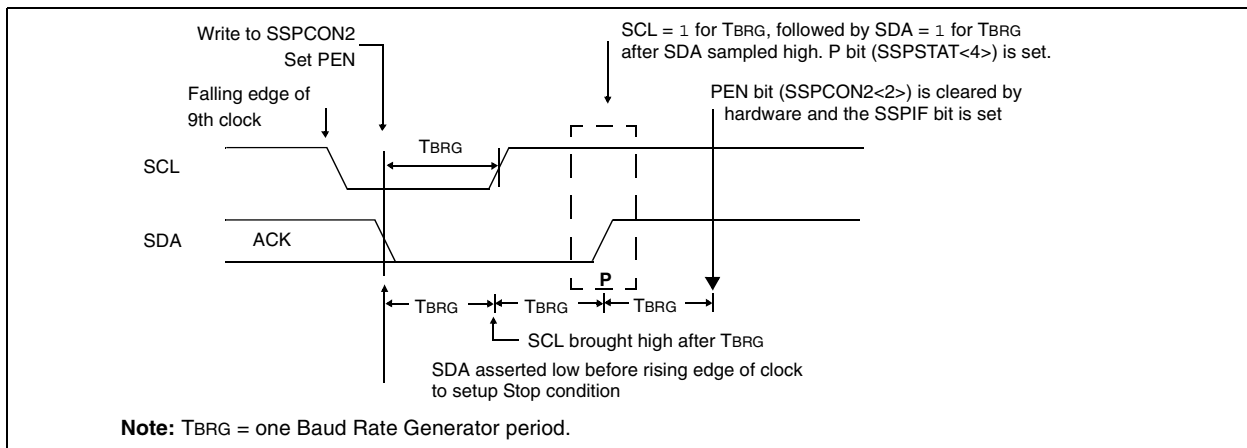


FIGURE 17-24: STOP CONDITION RECEIVE OR TRANSMIT MODE



PIC18FXX8

TABLE 18-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (Kbps)	Fosc = 40 MHz			33 MHz			25 MHz			20 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	9.60	-0.07	214	9.59	-0.15	162	9.62	+0.16	129
19.2	19.23	+0.16	129	19.28	+0.39	106	19.30	+0.47	80	19.23	+0.16	64
76.8	75.76	-1.36	32	76.39	-0.54	26	78.13	+1.73	19	78.13	+1.73	15
96	96.15	+0.16	25	98.21	+2.31	20	97.66	+1.73	15	96.15	+0.16	12
300	312.50	+4.17	7	294.64	-1.79	6	312.50	+4.17	4	312.50	+4.17	3
500	500	0	4	515.63	+3.13	3	520.83	+4.17	2	416.67	-16.67	2
HIGH	2500	-	0	2062.50	-	0	1562.50	-	0	1250	-	0
LOW	9.77	-	255	8.06	-	255	6.10	-	255	4.88	-	255

BAUD RATE (Kbps)	Fosc = 16 MHz			10 MHz			7.15909 MHz			5.0688 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	2.41	+0.23	185	2.40	0	131
9.6	9.62	+0.16	103	9.62	+0.16	64	9.52	-0.83	46	9.60	0	32
19.2	19.23	+0.16	51	18.94	-1.36	32	19.45	+1.32	22	18.64	-2.94	16
76.8	76.92	+0.16	12	78.13	+1.73	7	74.57	-2.90	5	79.20	+3.13	3
96	100	+4.17	9	89.29	-6.99	6	89.49	-6.78	4	105.60	+10.00	2
300	333.33	+11.11	2	312.50	+4.17	1	447.44	+49.15	0	316.80	+5.60	0
500	500	0	1	625	+25.00	0	447.44	-10.51	0	NA	-	-
HIGH	1000	-	0	625	-	0	447.44	-	0	316.80	-	0
LOW	3.91	-	255	2.44	-	255	1.75	-	255	1.24	-	255

BAUD RATE (Kbps)	Fosc = 4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	0.30	+0.16	207	0.29	-2.48	6
1.2	1.20	+0.16	207	1.20	+0.23	185	1.20	+0.16	51	1.02	-14.67	1
2.4	2.40	+0.16	103	2.41	+0.23	92	2.40	+0.16	25	2.05	-14.67	0
9.6	9.62	+0.16	25	9.73	+1.32	22	8.93	-6.99	6	NA	-	-
19.2	19.23	+0.16	12	18.64	-2.90	11	20.83	+8.51	2	NA	-	-
76.8	NA	-	-	74.57	-2.90	2	62.50	-18.62	0	NA	-	-
96	NA	-	-	111.86	+16.52	1	NA	-	-	NA	-	-
300	NA	-	-	223.72	-25.43	0	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	250	-	0	55.93	-	0	62.50	-	0	2.05	-	0
LOW	0.98	-	255	0.22	-	255	0.24	-	255	0.008	-	255

REGISTER 19-13: RXB1CON: RECEIVE BUFFER 1 CONTROL REGISTER

R/C-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0	R-0
RXFUL ⁽¹⁾	RXM1 ⁽¹⁾	RXM0 ⁽¹⁾	—	RXRTRRO	FILHIT2	FILHIT1	FILHIT0
bit 7				bit 0			

- bit 7 **RXFUL:** Receive Full Status bit⁽¹⁾
 1 = Receive buffer contains a received message
 0 = Receive buffer is open to receive a new message
Note: This bit is set by the CAN module and should be cleared by software after the buffer is read.
- bit 6-5 **RXM1:RXM0:** Receive Buffer Mode bits⁽¹⁾
 11 = Receive all messages (including those with errors)
 10 = Receive only valid messages with extended identifier
 01 = Receive only valid messages with standard identifier
 00 = Receive all valid messages
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **RXRTRRO:** Receive Remote Transfer Request bit (read-only)
 1 = Remote transfer request
 0 = No remote transfer request
- bit 2-0 **FILHIT2:FILHIT0:** Filter Hit bits
 These bits indicate which acceptance filter enabled the last message reception into Receive Buffer 1.
 111 = Reserved
 110 = Reserved
 101 = Acceptance Filter 5 (RXF5)
 100 = Acceptance Filter 4 (RXF4)
 011 = Acceptance Filter 3 (RXF3)
 010 = Acceptance Filter 2 (RXF2)
 001 = Acceptance Filter 1 (RXF1), only possible when RXB0DBEN bit is set
 000 = Acceptance Filter 0 (RXF0), only possible when RXB0DBEN bit is set
Note 1: Bits RXFUL, RXM1 and RXM0 of RXB1CON are not mirrored in RXB0CON.

Legend:

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

PIC18FXX8

REGISTER 19-23: RXFnEIDH: RECEIVE ACCEPTANCE FILTER n EXTENDED IDENTIFIER, HIGH BYTE REGISTERS

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8

bit 7

bit 0

bit 7-0

EID15:EID8: Extended Identifier Filter bits

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

REGISTER 19-24: RXFnEIDL: RECEIVE ACCEPTANCE FILTER n EXTENDED IDENTIFIER, LOW BYTE REGISTERS

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0

bit 7

bit 0

bit 7-0

EID7:EID0: Extended Identifier Filter bits

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

REGISTER 19-25: RXMnSIDH: RECEIVE ACCEPTANCE MASK n STANDARD IDENTIFIER MASK, HIGH BYTE REGISTERS

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3

bit 7

bit 0

bit 7-0

SID10:SID3: Standard Identifier Mask bits or Extended Identifier Mask bits EID28:EID21

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

PIC18FXX8

REGISTER 19-35: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
IRXIP	WAKIP	ERRIP	TXB2IP	TXB1IP	TXB0IP	RXB1IP	RXB0IP
bit 7						bit 0	

- bit 7 **IRXIP:** CAN Invalid Received Message Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 6 **WAKIP:** CAN bus Activity Wake-up Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 5 **ERRIP:** CAN bus Error Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 4 **TXB2IP:** CAN Transmit Buffer 2 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 3 **TXB1IP:** CAN Transmit Buffer 1 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 2 **TXB0IP:** CAN Transmit Buffer 0 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 1 **RXB1IP:** CAN Receive Buffer 1 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 0 **RXB0IP:** CAN Receive Buffer 0 Interrupt Priority bit
1 = High priority
0 = Low priority

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

19.12 Error Detection

The CAN protocol provides sophisticated error detection mechanisms. The following errors can be detected.

19.12.1 CRC ERROR

With the Cyclic Redundancy Check (CRC), the transmitter calculates special check bits for the bit sequence, from the start of a frame until the end of the data field. This CRC sequence is transmitted in the CRC field. The receiving node also calculates the CRC sequence using the same formula and performs a comparison to the received sequence. If a mismatch is detected, a CRC error has occurred and an error frame is generated. The message is repeated.

19.12.2 ACKNOWLEDGE ERROR

In the Acknowledge field of a message, the transmitter checks if the Acknowledge slot (which was sent out as a recessive bit) contains a dominant bit. If not, no other node has received the frame correctly. An Acknowledge Error has occurred; an error frame is generated and the message will have to be repeated.

19.12.3 FORM ERROR

If a node detects a dominant bit in one of the four segments, including end of frame, interframe space, Acknowledge delimiter or CRC delimiter, then a Form Error has occurred and an error frame is generated. The message is repeated.

19.12.4 BIT ERROR

A Bit Error occurs if a transmitter sends a dominant bit and detects a recessive bit, or if it sends a recessive bit and detects a dominant bit, when monitoring the actual bus level and comparing it to the just transmitted bit. In the case where the transmitter sends a recessive bit and a dominant bit is detected during the arbitration field and the Acknowledge slot, no Bit Error is generated because normal arbitration is occurring.

19.12.5 STUFF BIT ERROR

If, between the start of frame and the CRC delimiter, six consecutive bits with the same polarity are detected, the bit stuffing rule has been violated. A Stuff Bit Error occurs and an error frame is generated. The message is repeated.

19.12.6 ERROR STATES

Detected errors are made public to all other nodes via error frames. The transmission of the erroneous message is aborted and the frame is repeated as soon as possible. Furthermore, each CAN node is in one of the three error states “error-active”, “error-passive” or “bus-off” according to the value of the internal error counters. The error-active state is the usual state, where the bus node can transmit messages and activate error frames (made of dominant bits) without any restrictions. In the error-passive state, messages and passive error frames (made of recessive bits) may be transmitted. The bus-off state makes it temporarily impossible for the station to participate in the bus communication. During this state, messages can neither be received nor transmitted.

19.12.7 ERROR MODES AND ERROR COUNTERS

The PIC18FXX8 contains two error counters: the Receive Error Counter (RXERRCNT) and the Transmit Error Counter (TXERRCNT). The values of both counters can be read by the MCU. These counters are incremented or decremented in accordance with the CAN bus specification.

The PIC18FXX8 is error-active if both error counters are below the error-passive limit of 128. It is error-passive if at least one of the error counters equals or exceeds 128. It goes to bus-off if the transmit error counter equals or exceeds the bus-off limit of 256. The device remains in this state until the bus-off recovery sequence is received. The bus-off recovery sequence consists of 128 occurrences of 11 consecutive recessive bits (see Figure 19-10). Note that the CAN module, after going bus-off, will recover back to error-active without any intervention by the MCU if the bus remains Idle for 128 x 11 bit times. If this is not desired, the error Interrupt Service Routine should address this. The current error mode of the CAN module can be read by the MCU via the COMSTAT register.

Additionally, there is an Error State Warning flag bit, EWARN, which is set if at least one of the error counters equals or exceeds the error warning limit of 96. EWARN is reset if both error counters are less than the error warning limit.

REGISTER 24-7: CONFIG6L: CONFIGURATION REGISTER 6 LOW (BYTE ADDRESS 30000Ah)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	WRT3 ⁽¹⁾	WRT2 ⁽¹⁾	WRT1	WRT0

bit 7 bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **WRT3:** Write Protection bit⁽¹⁾

1 = Block 3 (006000-007FFFh) not write-protected

0 = Block 3 (006000-007FFFh) write-protected

bit 2 **WRT2:** Write Protection bit⁽¹⁾

1 = Block 2 (004000-005FFFh) not write-protected

0 = Block 2 (004000-005FFFh) write-protected

bit 1 **WRT1:** Write Protection bit

1 = Block 1 (002000-003FFFh) not write-protected

0 = Block 1 (002000-003FFFh) write-protected

bit 0 **WRT0:** Write Protection bit

1 = Block 0 (000200-001FFFh) not write-protected

0 = Block 0 (000200-001FFFh) write-protected

Note 1: Unimplemented in PIC18FX48 devices; maintain this bit set.

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

REGISTER 24-8: CONFIG6H: CONFIGURATION REGISTER 6 HIGH (BYTE ADDRESS 30000Bh)

R/P-1	R/P-1	R-1	U-0	U-0	U-0	U-0	U-0
WRTD	WRTB	WRTC	—	—	—	—	—

bit 7 bit 0

bit 7 **WRTD:** Data EEPROM Write Protection bit

1 = Data EEPROM not write-protected

0 = Data EEPROM write-protected

bit 6 **WRTB:** Boot Block Write Protection bit

1 = Boot Block (000000-0001FFh) not write-protected

0 = Boot Block (000000-0001FFh) write-protected

bit 5 **WRTC:** Configuration Register Write Protection bit

1 = Configuration registers (300000-3000FFh) not write-protected

0 = Configuration registers (300000-3000FFh) write-protected

Note: This bit is read-only and cannot be changed in user mode.

bit 4-0 **Unimplemented:** Read as '0'

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18FXX8

TABLE 25-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
			MSb		LSb			
LITERAL OPERATIONS								
ADDLW k	Add literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	AND literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW k	Inclusive OR literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR f, k	Move literal (12-bit) 2nd word to FSRx 1st word	2	1110	1110	00ff	kkkk	None	
			1111	0000	kkkk	kkkk		
MOVLB k	Move literal to BSR<3:0>	1	0000	0001	0000	kkkk	None	
MOVLW k	Move literal to WREG	1	0000	1110	kkkk	kkkk	None	
MULLW k	Multiply literal with WREG	1	0000	1101	kkkk	kkkk	None	
RETLW k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
SUBLW k	Subtract WREG from literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS								
TBLRD*	Table Read	2	0000	0000	0000	1000	None	
TBLRD*+	Table Read with post-increment		0000	0000	0000	1001	None	
TBLRD*-	Table Read with post-decrement		0000	0000	0000	1010	None	
TBLRD+*	Table Read with pre-increment		0000	0000	0000	1011	None	
TBLWT*	Table Write	2 (5)	0000	0000	0000	1100	None	
TBLWT*+	Table Write with post-increment		0000	0000	0000	1101	None	
TBLWT*-	Table Write with post-decrement		0000	0000	0000	1110	None	
TBLWT+*	Table Write with pre-increment		0000	0000	0000	1111	None	

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

CLRF		Clear f						
Syntax:	[<i>label</i>] CLRF f [,a]							
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]							
Operation:	000h → f 1 → Z							
Status Affected:	Z							
Encoding:	<table border="1"><tr><td>0110</td><td>101a</td><td>ffff</td><td>ffff</td></tr></table>				0110	101a	ffff	ffff
0110	101a	ffff	ffff					
Description:	Clears the contents of the specified register. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Process Data	Write register 'f'				

Example: CLRF FLAG_REG

Before Instruction
FLAG_REG = 0x5A

After Instruction
FLAG_REG = 0x00

CLRWDT		Clear Watchdog Timer							
Syntax:	[<i>label</i>] CLRWDT								
Operands:	None								
Operation:	000h → WDT, 000h → WDT postscaler, 1 → \overline{TO} , 1 → \overline{PD}								
Status Affected:	\overline{TO} , \overline{PD}								
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0100</td></tr></table>				0000	0000	0000	0100	
0000	0000	0000	0100						
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits \overline{TO} and \overline{PD} are set.								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	Q1	Q2	Q3	Q4					
	Decode	No operation	Process Data	No operation					

Example: CLRWDT

Before Instruction
WDT Counter = ?

After Instruction
WDT Counter = 0x00
WDT Postscaler = 0
 \overline{TO} = 1
 \overline{PD} = 1

FIGURE 27-19: MASTER SSP I²C™ BUS START/STOP BITS TIMING WAVEFORMS

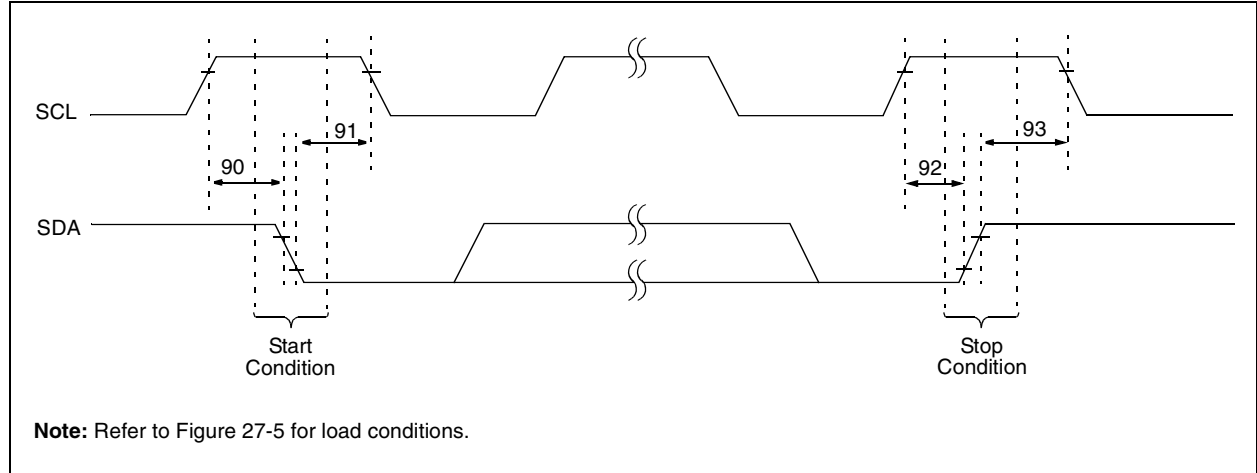


TABLE 27-19: MASTER SSP I²C™ BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	$2(T_{OSC})(BRG + 1)$	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	$2(T_{OSC})(BRG + 1)$	—		
			1 MHz mode ⁽¹⁾	$2(T_{OSC})(BRG + 1)$	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	$2(T_{OSC})(BRG + 1)$	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	$2(T_{OSC})(BRG + 1)$	—		
			1 MHz mode ⁽¹⁾	$2(T_{OSC})(BRG + 1)$	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	$2(T_{OSC})(BRG + 1)$	—	ns	
			400 kHz mode	$2(T_{OSC})(BRG + 1)$	—		
			1 MHz mode ⁽¹⁾	$2(T_{OSC})(BRG + 1)$	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	$2(T_{OSC})(BRG + 1)$	—	ns	
			400 kHz mode	$2(T_{OSC})(BRG + 1)$	—		
			1 MHz mode ⁽¹⁾	$2(T_{OSC})(BRG + 1)$	—		

Note 1: Maximum pin capacitance = 10 pF for all I²C™ pins.

FIGURE 27-20: MASTER SSP I²C™ BUS DATA TIMING

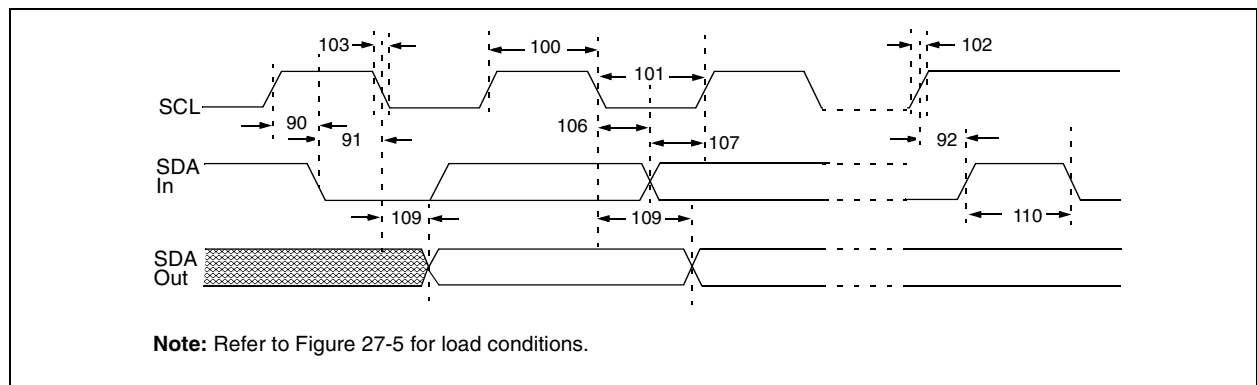


FIGURE 27-21: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

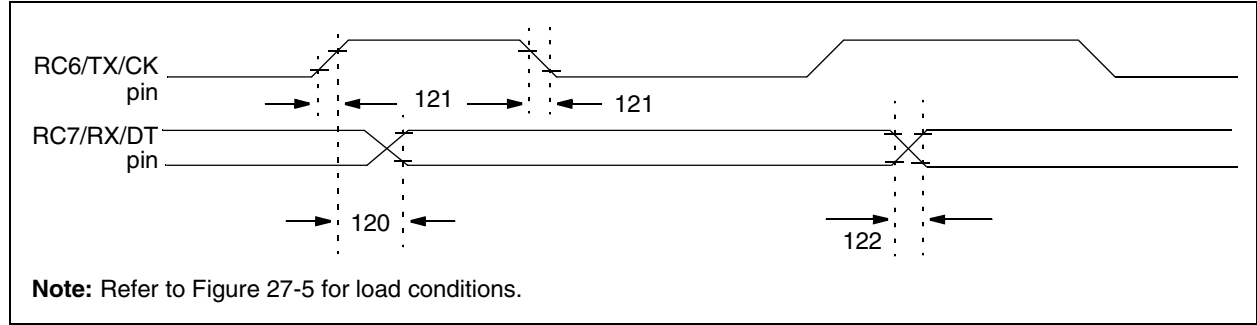


TABLE 27-21: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (Master & Slave) Clock High to Data-Out Valid	PIC18FXX8	—	50	ns
			PIC18LFXX8	—	150	ns
121	Tckrf	Clock Out Rise Time and Fall Time (Master mode)	PIC18FXX8	—	25	ns
			PIC18LFXX8	—	60	ns
122	Tdtrf	Data-Out Rise Time and Fall Time	PIC18FXX8	—	25	ns
			PIC18LFXX8	—	60	ns

FIGURE 27-22: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

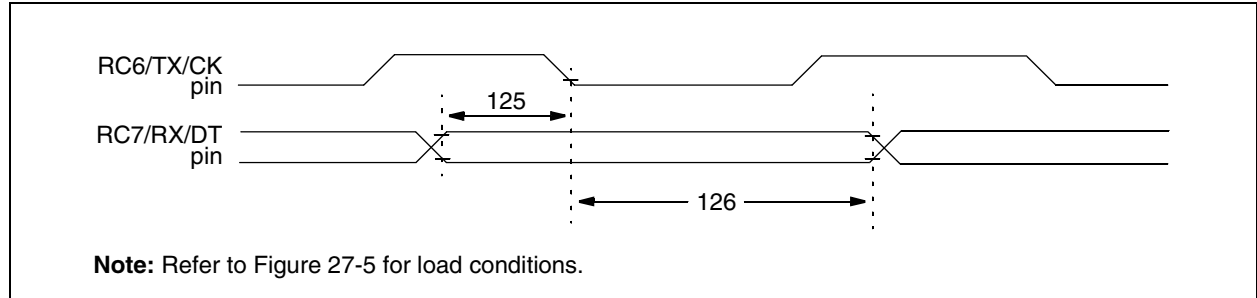


TABLE 27-22: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TdtV2ckl	SYNC RCV (Master & Slave) Data-Hold before CK ↓ (DT hold time)	10	—	ns	
126	TckL2dtl	Data-Hold after CK ↓ (DT hold time)	15	—	ns	

PIC18FXX8

FIGURE 28-3: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} (HS/PLL MODE)

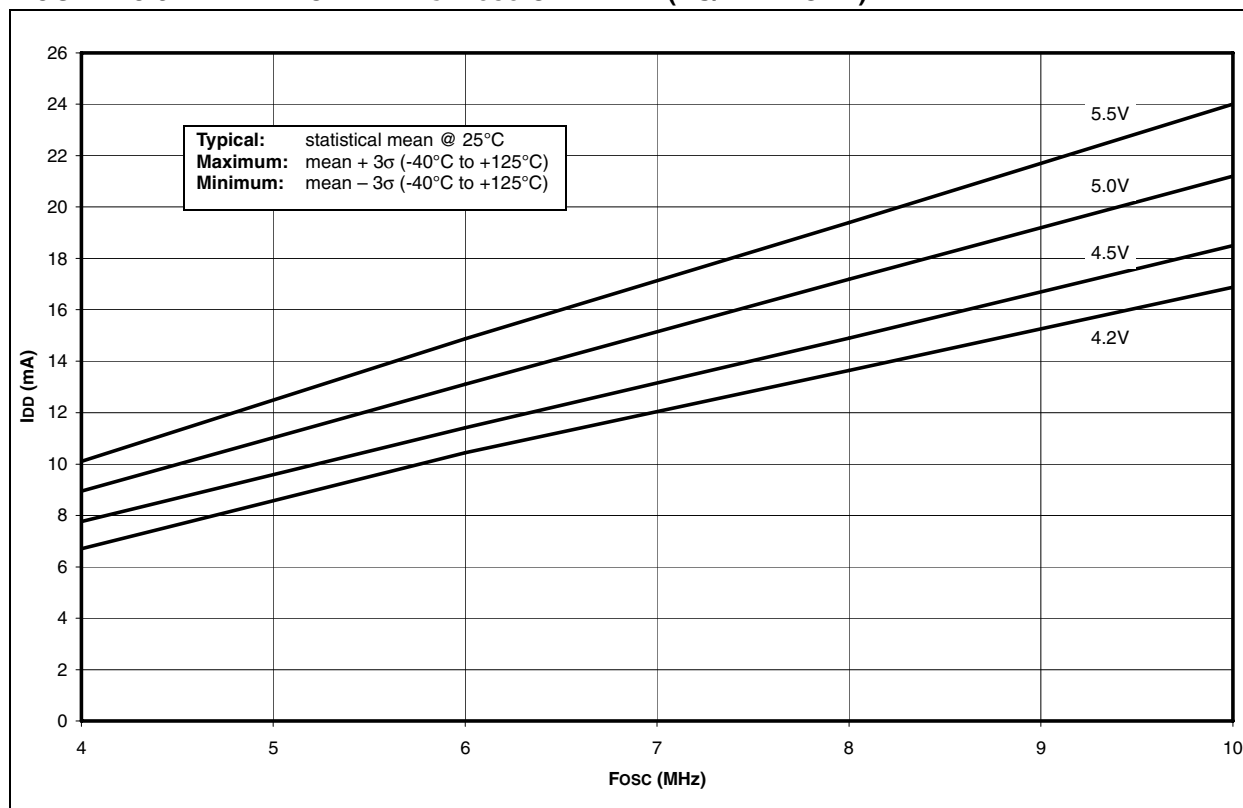
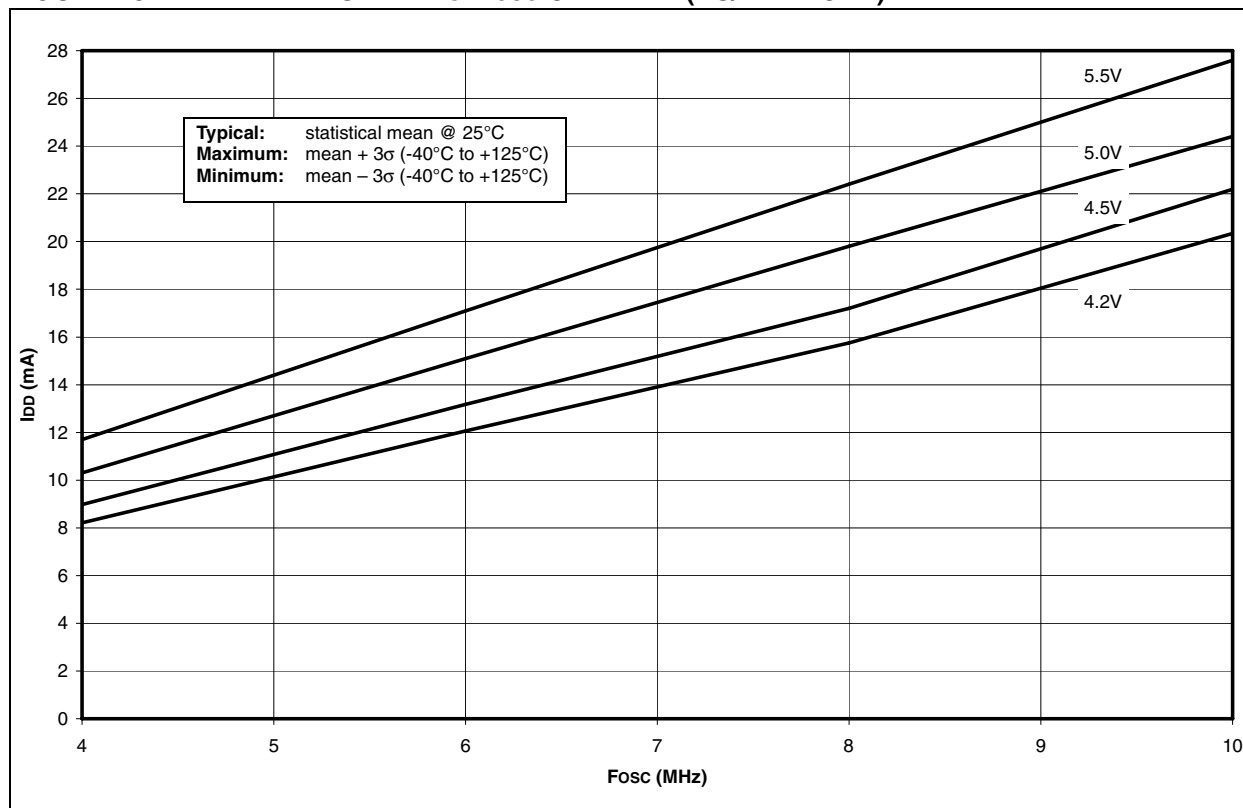


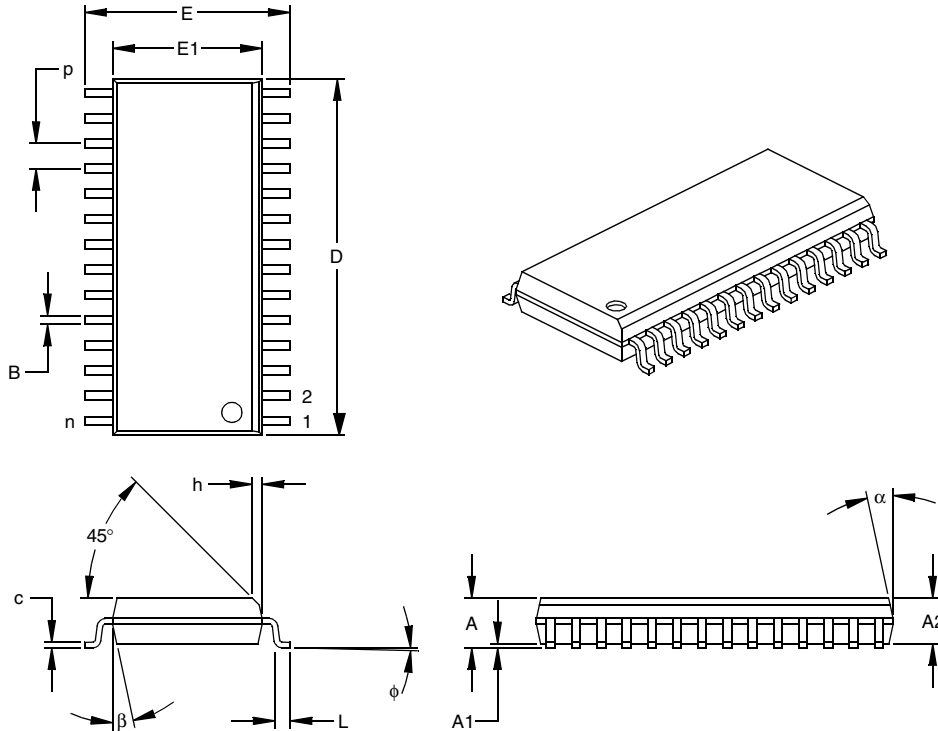
FIGURE 28-4: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} (HS/PLL MODE)



PIC18FXX8

28-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packageing>



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	28			28		
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.288	.295	.299	7.32	7.49	7.59
Overall Length	D	.695	.704	.712	17.65	17.87	18.08
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle Top	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.013	0.23	0.28	0.33
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-052

PIC18FXX8

RXBnSIDL (Receive Buffer n Standard Identifier, Low Byte).....	212
RXERRCNT (Receive Error Count).....	214
RXFnEIDH (Receive Acceptance Filter n Extended Identifier, High Byte).....	216
RXFnEIDL (Receive Acceptance Filter n Extended Identifier, Low Byte).....	216
RXFnSIDH (Receive Acceptance Filter n Standard Identifier Filter, High Byte).....	215
RXFnSIDL (Receive Acceptance Filter n Standard Identifier Filter, Low Byte).....	215
RXMnEIDH (Receive Acceptance Mask n Extended Identifier Mask, High Byte).....	217
RXMnEIDL (Receive Acceptance Mask n Extended Identifier Mask, Low Byte).....	217
RXMnSIDH (Receive Acceptance Mask n Standard Identifier Mask, High Byte).....	216
RXMnSIDL (Receive Acceptance Mask n Standard Identifier Mask, Low Byte).....	217
SSPCON1 (MSSP Control 1, I ² C Mode).....	154
SSPCON1 (MSSP Control 1, SPI Mode).....	145
SSPCON2 (MSSP Control 2, I ² C Mode).....	155
SSPSTAT (MSSP Status, I ² C Mode).....	153
SSPSTAT (MSSP Status, SPI Mode).....	144
Status.....	57
STKPTR (Stack Pointer).....	39
T0CON (Timer0 Control).....	109
T1CON (Timer1 Control).....	113
T2CON (Timer2 Control).....	117
T3CON (Timer3 Control).....	119
TRISE (PORTE Direction/PSP Control).....	105
TXBnCON (Transmit Buffer n Control).....	206
TXBnDLC (Transmit Buffer n Data Length Code).....	209
TXBnDm (Transmit Buffer n Data Field Byte m).....	208
TXBnEIDH (Transmit Buffer n Extended Identifier, High Byte).....	207
TXBnEIDL (Transmit Buffer n Extended Identifier, Low Byte).....	208
TXBnSIDH (Transmit Buffer n Standard Identifier, High Byte).....	207
TXBnSIDL (Transmit Buffer n Standard Identifier, Low Byte).....	207
TXERRCNT (Transmit Error Count).....	209
TXSTA (Transmit Status and Control).....	183
WDTCON (Watchdog Timer Control).....	272
RESET.....	311
Reset.....	25, 265
MCLR Reset During Normal Operation.....	25
MCLR Reset During Sleep.....	25
Power-on Reset (POR).....	25
Programmable Brown-out Reset (PBOR).....	25
RESET Instruction.....	25
Stack Full Reset.....	25
Stack Underflow Reset.....	25
Watchdog Timer (WDT) Reset.....	25
RETfie.....	312
RETLW.....	312
RETURN.....	313
Revision History.....	385
RLCF.....	313
RLNCF.....	314
RRCF.....	314
RRNCF.....	315

S

SCI. <i>See</i> USART.	
SCK Pin.....	143
SDI Pin.....	143
SDO Pin.....	143
Serial Clock (SCK) Pin.....	143
Serial Communication Interface. <i>See</i> USART.	
Serial Peripheral Interface. <i>See</i> SPI.	
SETF.....	315
Slave Select (\overline{SS}) Pin.....	143
Slave Select, \overline{SS} Pin.....	143
SLEEP.....	316
Sleep.....	265, 274
Software Simulator (MPLAB SIM).....	324
Software Simulator (MPLAB SIM30).....	324
Special Event Trigger. <i>See</i> Compare.	
Special Features of the CPU.....	265
Configuration Bits.....	265
Configuration Bits and Device IDs.....	265
Configuration Registers.....	266–271
Special Function Register Map.....	47
Special Function Registers.....	44
SPI Mode	
Associated Registers.....	151
Bus Mode Compatibility.....	151
Effects of a Reset.....	151
Master Mode.....	148
Master/Slave Connection.....	147
Registers.....	144
Serial Clock.....	143
Serial Data In (SDI) Pin.....	143
Serial Data Out (SDO) Pin.....	143
Slave Select.....	143
Slave Select Synchronization.....	149
Sleep Operation.....	151
SPI Clock.....	148
SSPBUF Register.....	148
SSPSR Register.....	148
SSPOV.....	173
SSPOV Status Flag.....	173
SSPSTAT Register	
R/W Bit.....	156, 157
SUBFWB.....	316
SUBLW.....	317
SUBWF.....	317
SUBWFB.....	318
SWAPF.....	318

T

Table Pointer Operations (table).....	68
TBLRD.....	319
TBLWT.....	320
Timer0.....	109
16-bit Mode Timer Reads and Writes.....	111
Associated Registers.....	111
Operation.....	111
Overflow Interrupt.....	111
Prescaler.....	111
Prescaler. <i>See</i> Prescaler, Timer0.	
Switching Prescaler Assignment.....	111