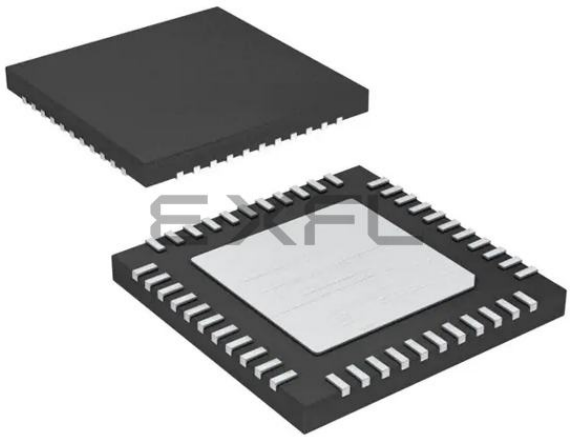


Welcome to E-XFL.COM

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"

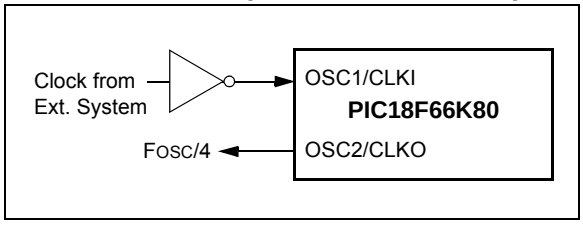
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f45k80-h-ml

3.5.2 EXTERNAL CLOCK INPUT (EC MODES)

The EC and ECPLL Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

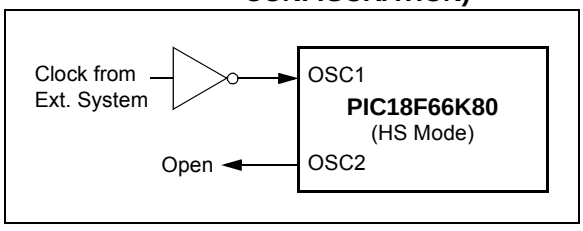
In the EC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 3-5 shows the pin connections for the EC Oscillator mode.

FIGURE 3-5: EXTERNAL CLOCK INPUT OPERATION (EC CONFIGURATION)



An external clock source may also be connected to the OSC1 pin in the HS mode, as shown in Figure 3-6. In this configuration, the divide-by-4 output on OSC2 is not available. Current consumption in this configuration will be somewhat higher than EC mode, as the internal oscillator's feedback circuitry will be enabled (in EC mode, the feedback circuit is disabled).

FIGURE 3-6: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)



3.5.3 PLL FREQUENCY MULTIPLIER

A Phase Lock Loop (PLL) circuit is provided as an option for users who want to use a lower frequency oscillator circuit or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals or users who require higher clock speeds from an internal oscillator.

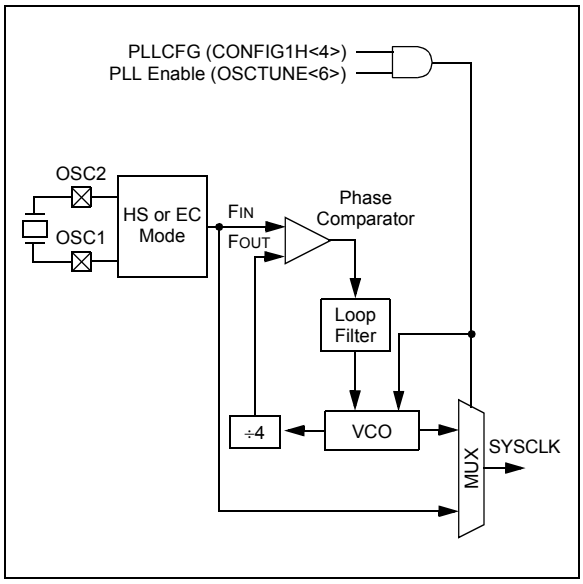
3.5.3.1 HSPLL and ECPLL Modes

The HSPLL and ECPLL modes provide the ability to selectively run the device at four times the external oscillating source to produce frequencies up to 64 MHz.

The PLL is enabled by setting the PLLEN bit (OSCTUNE<6>) or the PLLCFG bit (CONFIG1H<4>). For the HF-INTOSC as primary, the PLL must be enabled with the PLLEN. This provides a software control for the PLL, enabling even if PLLCFG is set to '1', so that the PLL is enabled only when the HF-INTOSC frequency is within the 4 MHz to16 MHz input range.

This also enables additional flexibility for controlling the application's clock speed in software. The PLLEN should be enabled in HF-INTOSC mode only if the input frequency is in the range of 4 MHz-16 MHz.

FIGURE 3-7: PLL BLOCK DIAGRAM



3.5.3.2 PLL and HF-INTOSC

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

The operation of INTOSC with the PLL is described in **Section 3.6.2 "INTPLL Modes"**. Care should be taken that the PLL is enabled only if the HF-INTOSC postscaler is configured for 4 MHz, 8 MHz or 16 MHz.

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'.

TABLE 6-2: PIC18F66K80 FAMILY REGISTER FILE SUMMARY (CONTINUED)

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR on page
E72h	RXFBCON1	CAN Buffer 3/2 Pointer Register								98
E71h	RXFBCON0	CAN Buffer 1/0 Pointer Register								98
E70h	SDFLC	—	—	—	CAN Device Net Count Register					98
E6Fh	RXF15EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	98
E6Eh	RXF15EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	98
E6Dh	RXF15SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	98
E6Ch	RXF15SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	98
E6Bh	RXF14EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	98
E6Ah	RXF14EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	98
E69h	RXF14SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	98
E68h	RXF14SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	98
E67h	RXF13EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	98
E66h	RXF13EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E65h	RXF13SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E64h	RXF13SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E63h	RXF12EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E62h	RXF12EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E61h	RXF12SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E60h	RXF12SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E5Fh	RXF11EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E5Eh	RXF11EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E5Dh	RXF11SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E5Ch	RXF11SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E5Bh	RXF10EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E5Ah	RXF10EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E59h	RXF10SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E58h	RXF10SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E57h	RXF9EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E56h	RXF9EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E55h	RXF9SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E54h	RXF9SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E53h	RXF8EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E52h	RXF8EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E51h	RXF8SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E50h	RXF8SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E4Fh	RXF7EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E4Eh	RXF7EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E4Dh	RXF7SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E4Ch	RXF7SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E4Bh	RXF6EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	99
E4Ah	RXF6EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	99
E49h	RXF6SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	99
E48h	RXF6SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	99
E47h	RXFCON1	CAN Receive Filter Control Register 1								99
E46h	RXFCON0	CAN Receive Filter Control Register 0								99
E45h	BRGCON3	WAKDIS	WAKFIL	—	—	—	SEG2PH2	SEG2PH1	SEG2PH0	99
E44h	BRGCON2	SEG2PHTS	SAM	SEG1PH2	SEG1PH1	SEG1PH0	PRSEG2	PRSEG1	PRSEG0	100
E43h	BRGCON1	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	100
E42h	TXERRCNT	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	100
E41h	RXERRCNT	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	100

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

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

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

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
1 = Accesses Flash program memory
0 = Accesses data EEPROM memory

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

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

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

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

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

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

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

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

EXAMPLE 8-1: DATA EEPROM READ

```
        MOVLW   DATA_EE_ADDRH      ;
        MOVWF   EEADRH             ; Upper bits of Data Memory Address to read
        MOVLW   DATA_EE_ADDR       ;
        MOVWF   EEADR              ; Lower bits of Data Memory Address to read
        BCF     EECON1, EEPGD      ; Point to DATA memory
        BCF     EECON1, CFGS       ; Access EEPROM
        BSF     EECON1, RD         ; EEPROM Read
        NOP
        MOVF    EEDATA, W          ; W = EEDATA
```

EXAMPLE 8-2: DATA EEPROM WRITE

```
                MOVLW   DATA_EE_ADDRH      ;
                MOVWF   EEADRH             ; Upper bits of Data Memory Address to write
                MOVLW   DATA_EE_ADDR       ;
                MOVWF   EEADR              ; Lower bits of Data Memory Address to write
                MOVLW   DATA_EE_DATA       ;
                MOVWF   EEDATA             ; Data Memory Value to write
                BCF     EECON1, EEPGD      ; Point to DATA memory
                BCF     EECON1, CFGS       ; Access EEPROM
                BSF     EECON1, WREN       ; Enable writes

                BCF     INTCON, GIE        ; Disable Interrupts
                MOVLW   55h                ;
Required        MOVWF   EECON2             ; Write 55h
Sequence        MOVLW   0AAh               ;
                MOVWF   EECON2             ; Write 0AAh
                BSF     EECON1, WR         ; Set WR bit to begin write
                BTFSC   EECON1, WR         ; Wait for write to complete GOTO $-2
                BSF     INTCON, GIE        ; Enable Interrupts

                                           ; User code execution
                BCF     EECON1, WREN       ; Disable writes on write complete (EEIF set)
```

TABLE 11-5: PORTC FUNCTIONS

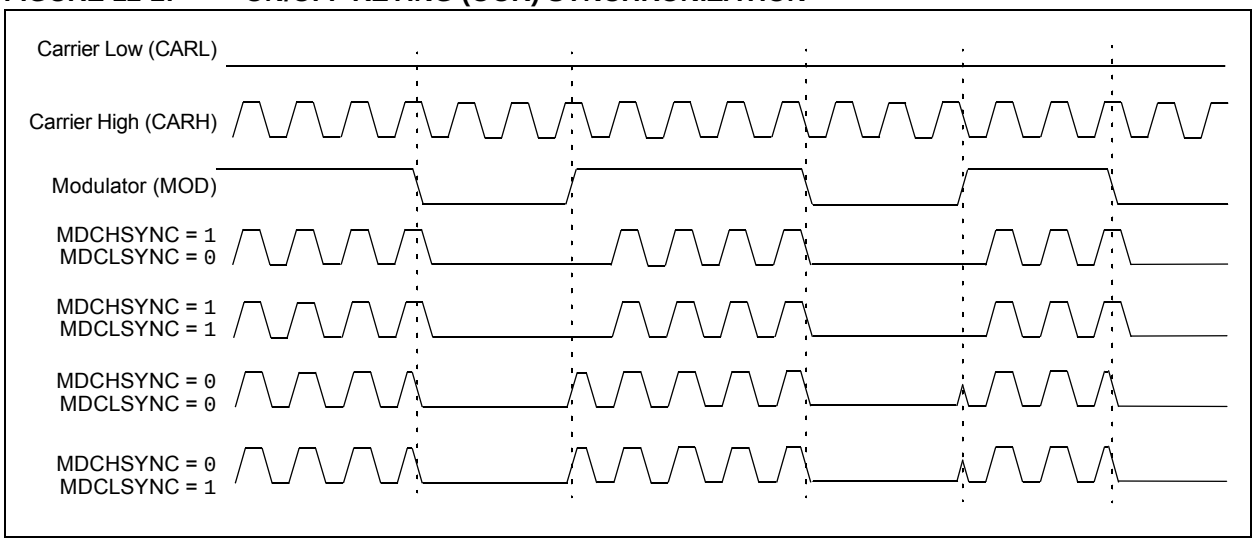
Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RC0/SOSCO/ SCLKI	RC0	0	O	DIG	LATC<0> data output.
		1	I	ST	PORTC<0> data input.
	SOSCO	1	I	ST	SOSC oscillator output.
	SCLKI	1	I	ST	Digital clock input; enabled when SOSC oscillator is disabled.
RC1/SOSCI	RC1	0	O	DIG	LATC<1> data output.
		1	I	ST	PORTC<1> data input.
	SOSCI	x	I	ANA	SOSC oscillator input.
RC2/T1G/ CCP2	RC2	0	O	DIG	LATC<2> data output.
		1	I	ST	PORTC<2> data input.
	T1G	x	I	ST	Timer1 external clock gate input.
	CCP2	0	O	DIG	CCP2 compare/PWM output; takes priority over port data.
		1	I	ST	CCP2 capture input.
RC3/REFO/ SCL/SCK	RC3	0	O	DIG	LATC<3> data output.
		1	I	ST	PORTC<3> data input.
	REFO	x	O	DIG	Reference output clock.
	SCL	0	O	DIG	I^2C™ clock output (MSSP module); takes priority over port data.
		1	I	I^2C	I^2C clock input (MSSP module); input type depends on module setting.
	SCK	0	O	DIG	SPI clock output (MSSP module); takes priority over port data.
		1	I	ST	SPI clock input (MSSP module).
RC4/SDA/SDI	RC4	0	O	DIG	LATC<4> data output.
		1	I	ST	PORTC<4> data input.
	SDA	1	O	DIG	I^2C data output (MSSP module); takes priority over port data.
		1	I	I^2C	I^2C data input (MSSP module); input type depends on module setting.
	SDI	1	I	ST	SPI data input (MSSP module).
RC5/SDO	RC5	0	O	DIG	LATC<5> data output.
		1	I	ST	PORTC<5> data input.
	SDO	0	O	DIG	SPI data output (MSSP module).
RC6/CANTX/ TX1/CK1/ CCP3	RC6	0	O	DIG	LATC<6> data output.
		1	I	ST	PORTC<6> data input.
	CANTX(2)	0	O	DIG	CAN bus TX.
	TX1(1)	0	O	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.
	CK1(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.
	CCP3	0	O	DIG	CCP3 compare/PWM output. Takes priority over port data.
		1	I	ST	CCP3 capture input.

Legend: O = Output; I = Input; I^2C = I^2C/SMBus; 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: The pin assignment for 28, 40 and 44-pin devices (PIC18F2XK80 and PIC18F4XK80).

2: The alternate pin assignment for CANRX and CANTX on 28, 40 and 44-pin devices (PIC18F4XK80) when the CANMX Configuration bit is set.

FIGURE 12-2: ON/OFF KEYING (OOK) SYNCHRONIZATION



EXAMPLE 12-1: NO SYNCHRONIZATION (MDCHSYNC = 0, MDCLSYNC = 0)

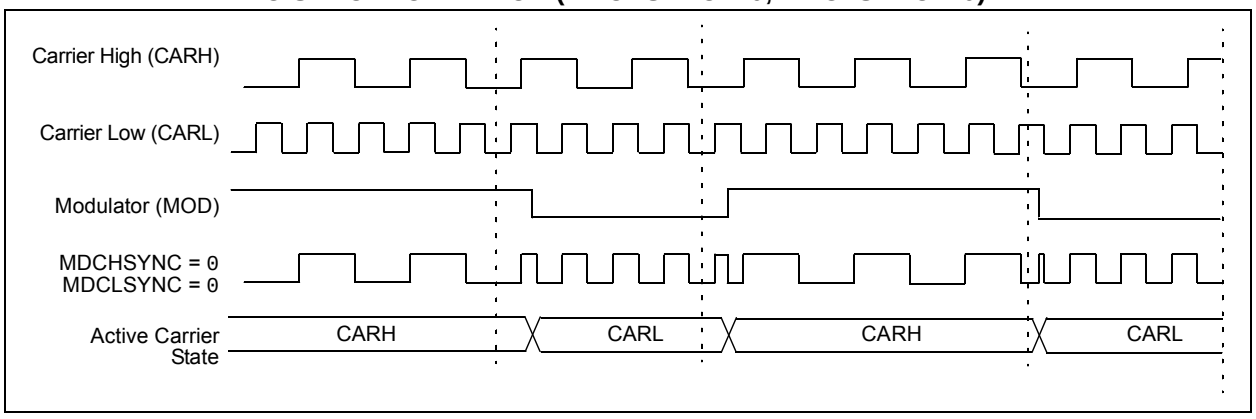


FIGURE 12-3: CARRIER HIGH SYNCHRONIZATION (MDCHSYNC = 1, MDCLSYNC = 0)

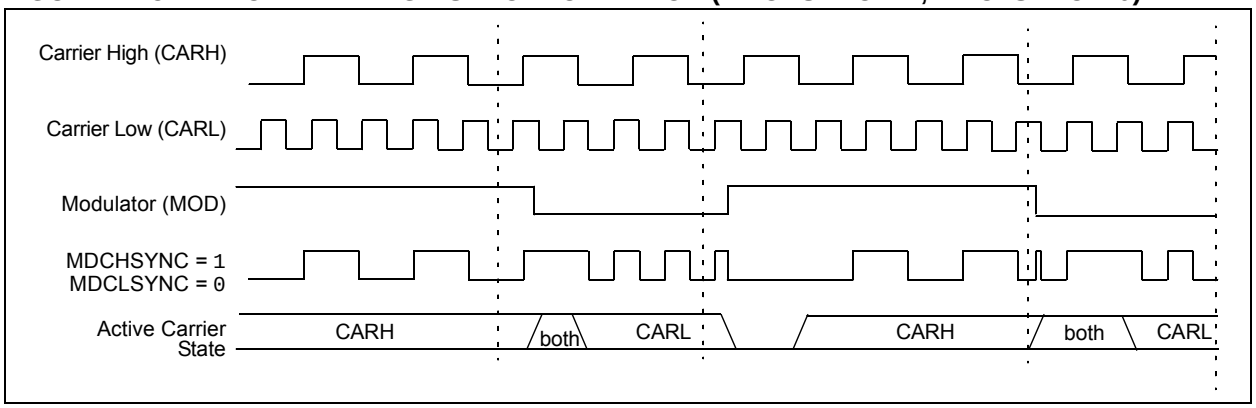


FIGURE 14-7: TIMER1 GATE SINGLE PULSE AND TOGGLE COMBINED MODE

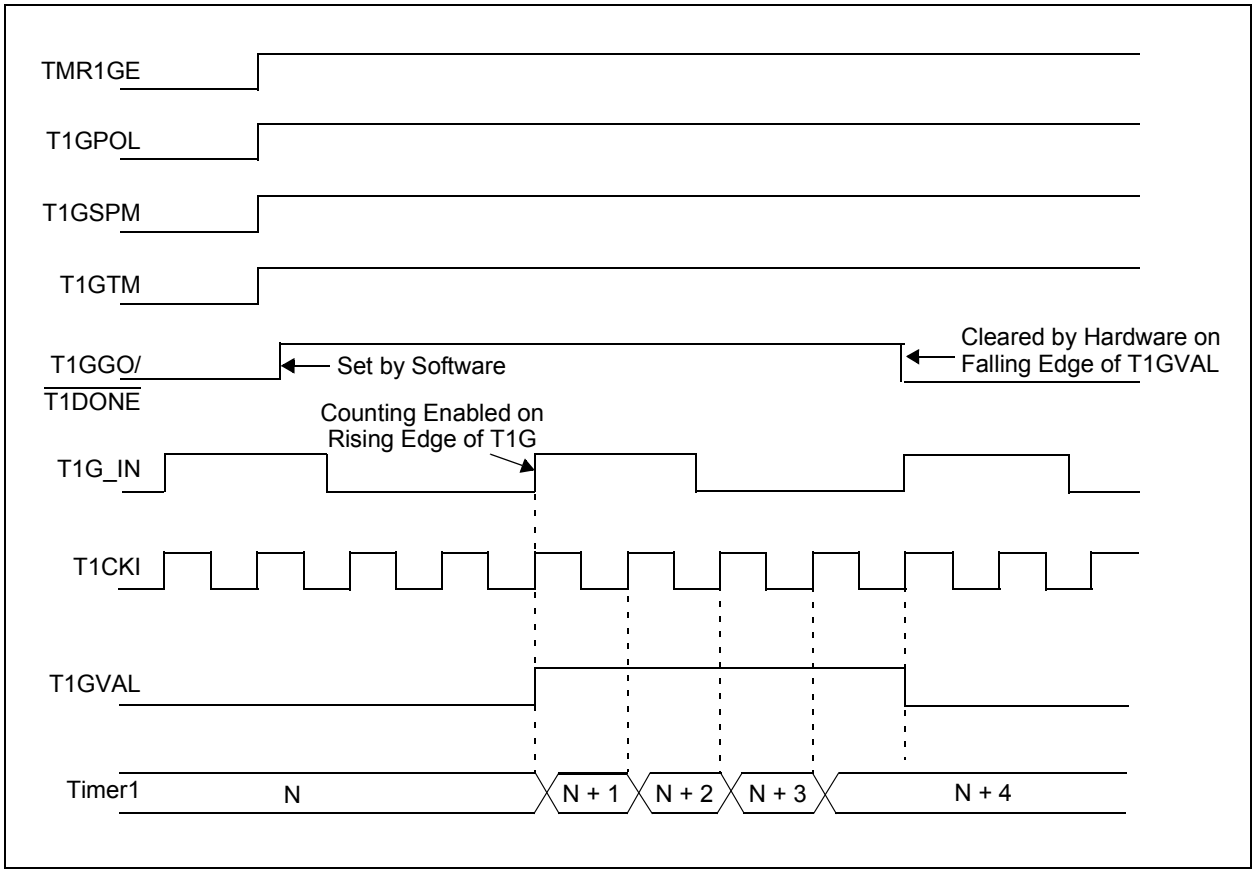


TABLE 14-5: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

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.

19.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE bit (PIE4<x>) clear to avoid false interrupts and should clear the flag bit, CCPxIF, following any such change in operating mode.

19.2.4 CCP PRESCALER

There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCPxM<3:0>). Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Doing that also will not clear the prescaler counter – meaning the first capture may be from a non-zero prescaler.

Example 19-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 19-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF   CCPxCON      ; Turn CCP module off
MOVLW  NEW_CAPT_PS  ; Load WREG with the
                    ; new prescaler mode
                    ; value and CCP ON
MOVWF  CCPxCON      ; Load CCPxCON with
                    ; this value
```

19.2.5 CAN MESSAGE TIME-STAMP (CCP2 ONLY)

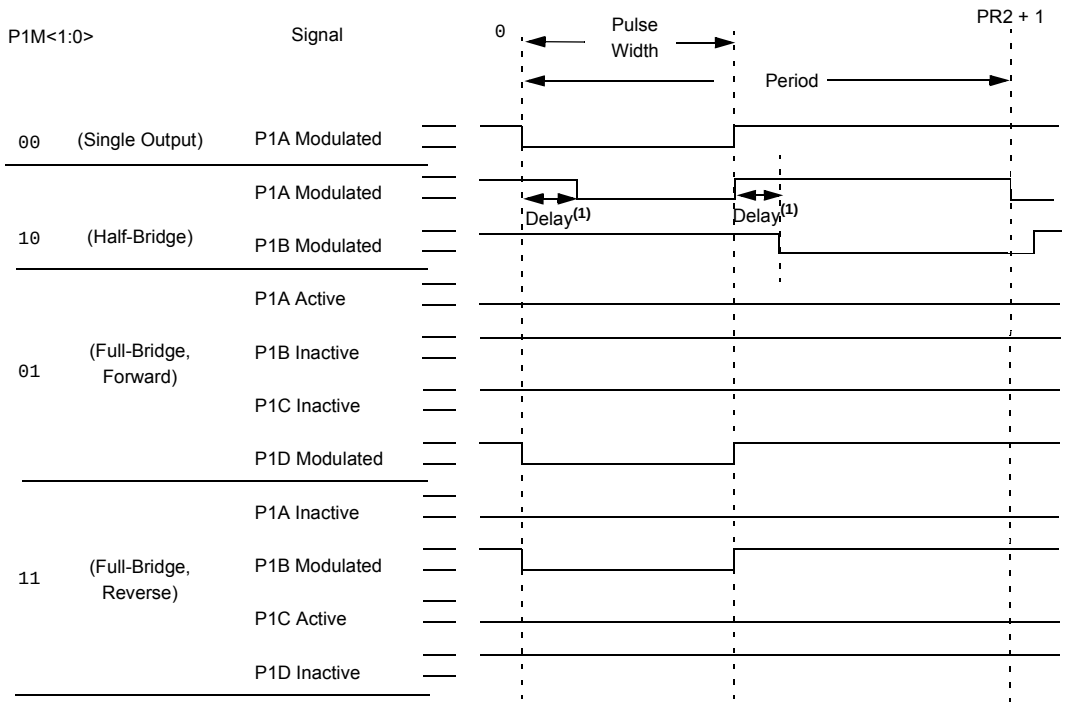
For CCP2, only the CAN capture event occurs when a message is received in any of the receive buffers. When configured, the CAN module provides the trigger to the CCP2 module to cause a capture event. This feature is provided to “time-stamp” the received CAN messages.

This feature is enabled by setting the CANCAP bit of the CAN I/O Control register (CIOCON<4>). The message receive signal from the CAN module then takes the place of the events on RC2/CCP2.

If this feature is selected, then four different capture options for CCP2M<3:0> are available:

- 0100 – Every time a CAN message is received
- 0101 – Every time a CAN message is received
- 0110 – Every 4th time a CAN message is received
- 0111 – Capture mode, every 16th time a CAN message is received

FIGURE 20-5: EXAMPLE ENHANCED PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW 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"**).

REGISTER 21-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
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 **WCOL:** Write Collision Detect bit
- 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit(1)

SPI Slave mode:
- 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
- 0 = No overflow

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(2)
- 1 = Enables the serial port and configures SCK, SDO, SDI and $\overline{SS}$ as serial port pins
- 0 = Disables the serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(3)
- 1010 = SPI Master mode: clock = FOSC/8
- 0101 = SPI Slave mode: clock = SCK pin; $\overline{SS}$ pin control disabled; $\overline{SS}$ can be used as I/O pin
- 0100 = SPI Slave mode: clock = SCK pin; $\overline{SS}$ pin control enabled
- 0011 = SPI Master mode: clock = TMR2 output/2
- 0010 = SPI Master mode: clock = FOSC/64
- 0001 = SPI Master mode: clock = FOSC/16
- 0000 = SPI Master mode: clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

2: When enabled, these pins must be properly configured as inputs or outputs.

3: Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.

21.4.2 OPERATION

The MSSP module functions are enabled by setting the MSSP Enable bit, SSPEN (SSPCON1<5>).

The SSPCON1 register allows control of the I^2C operation. Four mode selection bits (SSPCON1<3:0>) allow one of the following I^2C modes to be selected:

- I^2C Master mode, clock
- I^2C Slave mode (7-bit address)
- I^2C Slave mode (10-bit address)
- I^2C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I^2C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I^2C Firmware Controlled Master mode, slave is Idle

Selection of any I^2C mode with the SSPEN bit set forces the SCL and SDA pins to be open-drain, provided these pins are programmed as inputs by setting the appropriate TRISC bit. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

21.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

The I^2C Slave mode hardware will always generate an interrupt on an address match. Address masking will allow the hardware to generate an interrupt for more than one address (up to 31 in 7-bit addressing and up to 63 in 10-bit addressing). Through the mode select bits, the user can also choose to interrupt on Start and Stop bits.

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ($\overline{ACK}$) pulse and load the SSPBUF register with the received value currently in the SSPSR register.

Any combination of the following conditions will cause the MSSP module not to give this $\overline{ACK}$ pulse:

- The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- The overflow bit, SSPOV (SSPCON1<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit, SSPIF, is set. The BF bit is cleared by reading the SSPBUF register, while bit, SSPOV, is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I^2C specification, as well as the requirement of the MSSP module, are shown in timing Parameter 100 and Parameter 101.

21.4.3.1 Addressing

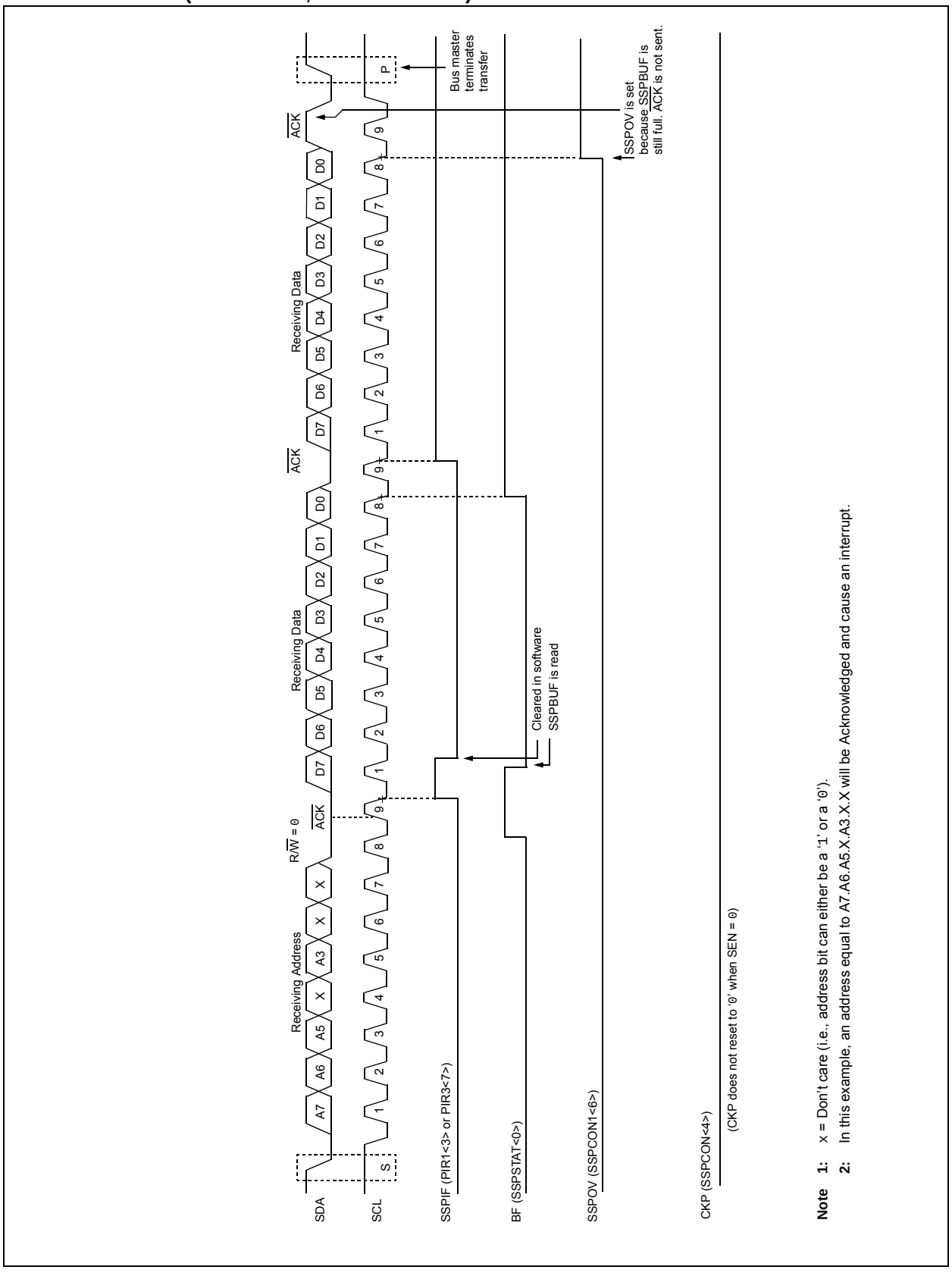
Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register, SSPSR<7:1>, is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An $\overline{ACK}$ pulse is generated.
4. The MSSP Interrupt Flag bit, SSPIF, is set (and interrupt is generated, if enabled) on the falling edge of the ninth SCL pulse.

In 10-Bit Addressing mode, two address bytes need to be received by the slave. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. The R/$\overline{W}$ (SSPSTAT<2>) bit must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSbs of the address. The sequence of events for 10-bit addressing is as follows, with Steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits, SSPIF, BF and UA, are set on address match).
2. Update the SSPADD register with second (low) byte of address (clears bit, UA, and releases the SCL line).
3. Read the SSPBUF register (clears bit, BF) and clear flag bit, SSPIF.
4. Receive second (low) byte of address (bits, SSPIF, BF and UA, are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit, UA.
6. Read the SSPBUF register (clears bit, BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits, SSPIF and BF, are set).
9. Read the SSPBUF register (clears bit, BF) and clear flag bit, SSPIF.

FIGURE 21-9: I²C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01011 (RECEPTION, 7-BIT ADDRESS)



Note 1: x = Don't care (i.e., address bit can either be a '1' or a '0').

2: In this example, an address equal to A7.A6.A5.X.A3.X.X will be Acknowledged and cause an interrupt.

21.4.4 CLOCK STRETCHING

Both 7-Bit and 10-Bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCL pin to be held low at the end of each data receive sequence.

21.4.4.1 Clock Stretching for 7-Bit Slave Receive Mode (SEN = 1)

In 7-Bit Slave Receive mode, on the falling edge of the ninth clock at the end of the ACK sequence, if the BF bit is set, the CKP bit in the SSPCON1 register is automatically cleared, forcing the SCL output to be held low. The CKP bit being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 21-15).

Note 1: If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software, regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

21.4.4.2 Clock Stretching for 10-Bit Slave Receive Mode (SEN = 1)

In 10-Bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address with the R/$\overline{W}$ bit cleared to '0'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

Note: If the user polls the UA bit and clears it by updating the SSPADD register before the falling edge of the ninth clock occurs, and if the user hasn't cleared the BF bit by reading the SSPBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

21.4.4.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 7-Bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 21-10).

Note 1: If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software regardless of the state of the BF bit.

21.4.4.4 Clock Stretching for 10-Bit Slave Transmit Mode

In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/$\overline{W}$ bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-Bit Slave Transmit mode (see Figure 21-13).

REGISTER 23-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
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 **ADFM:** A/D Result Format Select bit
1 = Right justified
0 = Left justified

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

bit 5-3 **ACQT<2:0>:** A/D Acquisition Time Select bits
111 = 20 TAD
110 = 16 TAD
101 = 12 TAD
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD(1)

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits
111 = FRC (clock derived from A/D RC oscillator)(1)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock derived from A/D RC oscillator)(1)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

REGISTER 27-1: CANCON: CAN CONTROL REGISTER

Mode 0	R/W-1	R/W-0	R/W-0	R/S-0	R/W-0	R/W-0	R/W-0	U-0
	REQOP2	REQOP1	REQOP0	ABAT	WIN2	WIN1	WIN0	—
Mode 1	R/W-1	R/W-0	R/W-0	R/S-0	U0	U-0	U-0	U-0
	REQOP2	REQOP1	REQOP0	ABAT	—	—	—	—
Mode 2	R/W-1	R/W-0	R/W-0	R/S-0	R-0	R-0	R-0	R-0
	REQOP2	REQOP1	REQOP0	ABAT	FP3	FP2	FP1	FP0
	bit 7							bit 0

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

bit 7-5 **REQOP<2:0>:** Request CAN Operation Mode bits
1xx = Requests Configuration mode
011 = Requests Listen Only mode
010 = Requests Loopback mode
001 = Disabled/Sleep mode
000 = Requests Normal mode

bit 4 **ABAT:** Abort All Pending Transmissions bit
1 = Abort all pending transmissions (in all transmit buffers)(1)
0 = Transmissions proceeding as normal

bit 3-1 Mode 0:
WIN<2:0>: Window Address bits
These bits select which of the CAN buffers to switch into the Access Bank area. This allows access to the buffer registers from any data memory bank. After a frame has caused an interrupt, the ICODE<3:0> bits can be copied to the WIN<2:0> bits to select the correct buffer. See Example 27-2 for a code example.
111 = Receive Buffer 0
110 = Receive Buffer 0
101 = Receive Buffer 1
100 = Transmit Buffer 0
011 = Transmit Buffer 1
010 = Transmit Buffer 2
001 = Receive Buffer 0
000 = Receive Buffer 0

bit 0 Mode 0:
Unimplemented: Read as '0'

bit 4-0 Mode 1:
Unimplemented: Read as '0'
Mode 2:
FP<3:0>: FIFO Read Pointer bits
These bits point to the message buffer to be read.
0000 = Receive Message Buffer 0
0001 = Receive Message Buffer 1
0010 = Receive Message Buffer 2
0011 = Receive Message Buffer 3
0100 = Receive Message Buffer 4
0101 = Receive Message Buffer 5
0110 = Receive Message Buffer 6
0111 = Receive Message Buffer 7
1000:1111 Reserved

Note 1: This bit will clear when all transmissions are aborted.

EXAMPLE 27-1: CHANGING TO CONFIGURATION MODE

```
    ; Request Configuration mode.
    MOVLW   B'10000000'                 ; Set to Configuration Mode.
    MOVWF   CANCON
    ; A request to switch to Configuration mode may not be immediately honored.
    ; Module will wait for CAN bus to be idle before switching to Configuration Mode.
    ; Request for other modes such as Loopback, Disable etc. may be honored immediately.
    ; It is always good practice to wait and verify before continuing.
ConfigWait:
    MOVF    CANSTAT, W                  ; Read current mode state.
    ANDLW   B'10000000'                 ; Interested in OPMODE bits only.
    TSTFSZ  WREG                        ; Is it Configuration mode yet?
    BRA     ConfigWait                  ; No. Continue to wait...
    ; Module is in Configuration mode now.
    ; Modify configuration registers as required.
    ; Switch back to Normal mode to be able to communicate.
```

EXAMPLE 27-2: WIN AND ICODE BITS USAGE IN INTERRUPT SERVICE ROUTINE TO ACCESS TX/RX BUFFERS

```
    ; Save application required context.
    ; Poll interrupt flags and determine source of interrupt
    ; This was found to be CAN interrupt
    ; TempCANCON and TempCANSTAT are variables defined in Access Bank low
    MOVFF   CANCON, TempCANCON          ; Save CANCON.WIN bits
                                        ; This is required to prevent CANCON
                                        ; from corrupting CAN buffer access
                                        ; in-progress while this interrupt
                                        ; occurred
    MOVFF   CANSTAT, TempCANSTAT        ; Save CANSTAT register
                                        ; This is required to make sure that
                                        ; we use same CANSTAT value rather
                                        ; than one changed by another CAN
                                        ; interrupt.
    MOVF    TempCANSTAT, W              ; Retrieve ICODE bits
    ANDLW   B'00001110'
    ADDWF   PCL, F                      ; Perform computed GOTO
                                        ; to corresponding interrupt cause
    BRA     NoInterrupt                 ; 000 = No interrupt
    BRA     ErrorInterrupt              ; 001 = Error interrupt
    BRA     TXB2Interrupt               ; 010 = TXB2 interrupt
    BRA     TXB1Interrupt               ; 011 = TXB1 interrupt
    BRA     TXB0Interrupt               ; 100 = TXB0 interrupt
    BRA     RXB1Interrupt               ; 101 = RXB1 interrupt
    BRA     RXB0Interrupt               ; 110 = RXB0 interrupt
                                        ; 111 = Wake-up on interrupt
WakeupInterrupt
    BCF     PIR3, WAKIF                 ; Clear the interrupt flag
    ;
    ; User code to handle wake-up procedure
    ;
    ;
    ; Continue checking for other interrupt source or return from here
    …
NoInterrupt
    …                                   ; PC should never vector here. User may
                                        ; place a trap such as infinite loop or pin/port
                                        ; indication to catch this error.
```

REGISTER 27-35: BnDLC: TX/RX BUFFER 'n' DATA LENGTH CODE REGISTERS IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL<n>) = 1](1)

U-0	R/W-x	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0
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 **TXRTR:** Transmitter Remote Transmission Request bit
1 = Transmitted message will have the RTR bit set
0 = Transmitted message will have the RTR bit cleared

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **DLC<3:0>:** Data Length Code bits
1111-1001 = Reserved
1000 = Data length = 8 bytes
0111 = Data length = 7 bytes
0110 = Data length = 6 bytes
0101 = Data length = 5 bytes
0100 = Data length = 4 bytes
0011 = Data length = 3 bytes
0010 = Data length = 2 bytes
0001 = Data length = 1 byte
0000 = Data length = 0 bytes

Note 1: These registers are available in Mode 1 and 2 only.

REGISTER 27-36: BSEL0: BUFFER SELECT REGISTER 0(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
B5TXEN	B4TXEN	B3TXEN	B2TXEN	B1TXEN	B0TXEN	—	—
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 7-2 **B<5:0>TXEN:** Buffer 5 to Buffer 0 Transmit Enable bits
1 = Buffer is configured in Transmit mode
0 = Buffer is configured in Receive mode

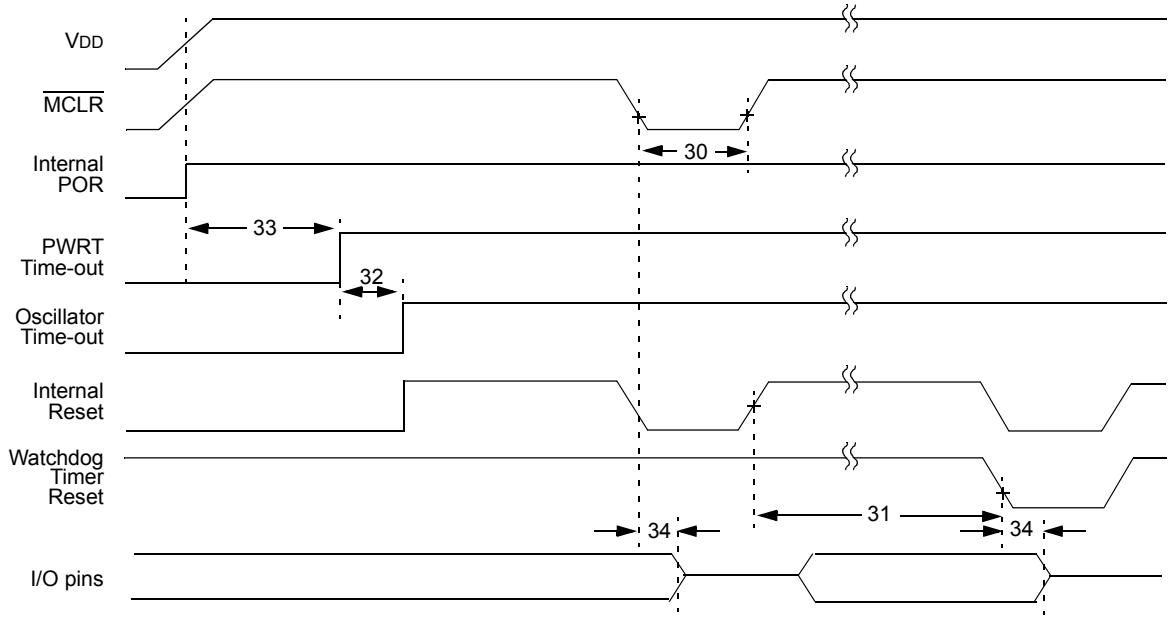
bit 1-0 **Unimplemented:** Read as '0'

Note 1: These registers are available in Mode 1 and 2 only.

TABLE 31-10: CLKO AND I/O TIMING REQUIREMENTS

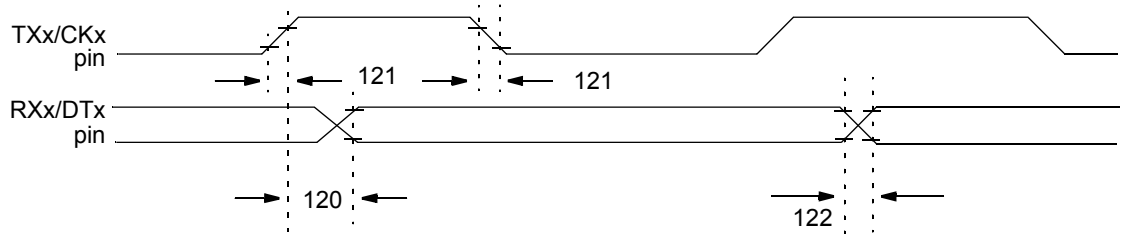
Param. No	Symbol	Characteristics	Min	Typ	Max	Units
150	TadV2alL	Address Out Valid to ALE ↓ (address setup time)	0.25 TCY – 10	—	—	ns
151	TalL2adl	ALE ↓ to Address Out Invalid (address hold time)	5	—	—	ns
155	TalL2oeL	ALE ↓ to $\overline{OE}$ ↓	10	0.125 TCY	—	ns
160	TadZ2oeL	AD High-Z to $\overline{OE}$ ↓ (bus release to $\overline{OE}$)	0	—	—	ns
161	ToeH2adD	$\overline{OE}$ ↑ to AD Driven	0.125 TCY – 5	—	—	ns
162	TadV2oeH	LS Data Valid before $\overline{OE}$ ↑ (data setup time)	20	—	—	ns
163	ToeH2adl	$\overline{OE}$ ↑ to Data In Invalid (data hold time)	0	—	—	ns
164	TalH2alL	ALE Pulse Width	—	0.25 TCY	—	ns
165	ToeL2oeH	$\overline{OE}$ Pulse Width	0.5 TCY – 5	0.5 TCY	—	ns
166	TalH2alH	ALE ↑ to ALE ↑ (cycle time)	—	TCY	—	ns
167	Tacc	Address Valid to Data Valid	0.75 TCY – 25	—	—	ns
168	Toe	$\overline{OE}$ ↓ to Data Valid		—	0.5 TCY – 25	ns
169	TalL2oeH	ALE ↓ to $\overline{OE}$ ↑	0.625 TCY – 10	—	0.625 TCY + 10	ns
171	TalH2csL	Chip Enable Active to ALE ↓	0.25 TCY – 20	—	—	ns
171A	TubL2oeH	AD Valid to Chip Enable Active	—	—	10	ns

FIGURE 31-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 31-3 for load conditions.

FIGURE 31-19: EUSARTx SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

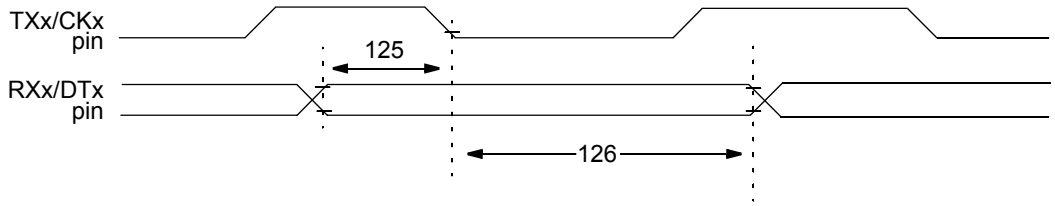


Note: Refer to Figure 31-3 for load conditions.

TABLE 31-23: EUSART/AUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
120	TCKH2DTV	SYNC XMIT (MASTER and SLAVE) Clock High to Data Out Valid	—	40	ns	
121	TCKRF	Clock Out Rise Time and Fall Time (Master mode)	—	20	ns	
122	TDTRF	Data Out Rise Time and Fall Time	—	20	ns	

FIGURE 31-20: EUSART/AUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 31-3 for load conditions.

TABLE 31-24: EUSART/AUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	SYNC RCV (MASTER and SLAVE) Data Hold before CKx ↓ (DTx hold time)	10	—	ns	
126	TCKL2DTL	Data Hold after CKx ↓ (DTx hold time)	15	—	ns	