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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	43
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 39x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19185-e-pt

TABLE 1-2: PIC16(L)F19155/56 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB3/C1IN2-/C2IN2-/IOCB3/ANB3/SEG11/COM6/ SEGCFLY2	RB3	TTL/ST	CMOS/OD	General purpose I/O.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
	IOCB3	TTL/ST	—	Interrupt-on-change input.
	ANB3	AN	—	ADC Channel input.
	SEG11	—	AN	LCD Analog output.
	COM6	—	AN	LCD Driver Common Outputs.
	SEGCFLY2	AN	—	LCD Drive Charge Pump Capacitor Inputs
RB4/ADCACT(1)/IOCB4/ANB4/COM0	RB4	TTL/ST	CMOS/OD	General purpose I/O.
	ADCACT(1)	TTL/ST	—	ADC Auto-Conversion Trigger input
	IOCB4	TTL/ST	—	Interrupt-on-change input.
	ANB4	AN	—	ADC Channel input.
	COM0	—	AN	LCD Driver Common Outputs.
RB5/T1G(1)/IOCB5/ANB5/SEG13/COM1	RB5	TTL/ST	CMOS/OD	General purpose I/O.
	T1G(1)	—	—	Timer1 Gate input.
	IOCB5	TTL/ST	—	Interrupt-on-change input.
	ANB5	AN	—	ADC Channel input.
	SEG13	—	AN	LCD Analog output.
	COM1	—	AN	LCD Driver Common Outputs.
RB6/CK2(3)/TX2(1)/CLCIN2(1)/IOCB6/ANB6/SEG14/ ICSPCLK	RB6	TTL/ST	CMOS/OD	General purpose I/O.
	CK2(3)	—	—	EUSART synchronous clock out
	TX2(1)	—	—	EUSART asynchronous TX data out
	CLCIN2(1)	—	—	Configurable Logic Cell source input.
	IOCB6	TTL/ST	—	Interrupt-on-change input.
	ANB6	AN	—	ADC Channel input.
	SEG14	—	AN	LCD Analog output.
	ICSPCLK	ST	—	In-Circuit Serial Programming™ and debugging clock input.
RB7/DK2(3)/RX2(1)/CLCIN3(1)/IOCB7/ANB7/SEG15/ DAC1OUT2/ICSPDAT	RB7	TTL/ST	CMOS/OD	General purpose I/O.
	DK2(3)	—	—	EUSART synchronous data output
	RX2(1)	—	—	EUSART receive input.
	CLCIN3(1)	—	—	Configurable Logic Cell source input.
	IOCB7	TTL/ST	—	Interrupt-on-change input.
	ANB7	AN	—	ADC Channel input.
	SEG15	—	AN	LCD Analog output.
	DAC1OUT2	—	AN	Digital-to-Analog Converter output.
	ICSPDAT	TTL/ST	TTL/ST	In-Circuit Serial Programming™ and debugging data input/output.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open-Drain; TTL = TTL compatible input‘; ST = Schmitt Trigger input with CMOS levels; I^2C = Schmitt Trigger input with I^2C; HV = High Voltage; XTAL = Crystal levels

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 14-2 for details on which PORT pins may be used for this signal.

2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 14-3.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins are configured for I^2C logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19155/56/75/76/85/86 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 14											
CPU CORE REGISTERS; see Table 4-3 for specifics											
70Ch	PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	--00 ---1	00-- ---1
70Dh	PIR1	OSFIF	CSWIF	—	—	—	—	ADTIF	ADIF	00-- --00	00-- --00
70Eh	PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	0000 0000	0000 0000
70Fh	PIR3	RC2IF	TX2IF	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	0000 --00	0000 --00
710h	PIR4	—	—	—	—	TMR4IF	—	TMR2IF	TMR1IF	0000 0000	0000 0000
711h	PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	0000 ---0	0000 ---0
712h	PIR6	CRIF	—	—	—	—	—	CCP2IF	CCP1IF	0--- --00	0--- --00
713h	PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	0000 0000	0000 0000
714h	PIR8	LCDIF	RTCCIF	—	—	—	SMT1PWAIF	SMT1PRAIF	SMT1IF	00-- -000	00-- -000
715h	—	Unimplemented								---- ----	---- ----
716h	PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	0000 0000	0000 0000
717h	PIE1	OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE	00-- --00	00-- --00
718h	PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	0000 0000	0000 0000
719h	PIE3	RC2IE	TX2IE	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	0000 --00	0000 --00
71Ah	PIE4	—	—	—	—	TMR4IF	—	TMR2IE	TMR1IE	0000 0000	0000 0000
71Bh	PIE5	CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	0000 ---0	0000 ---0
71Ch	PIE6	CRIE	—	—	—	—	—	CCP2IE	CCP1IE	0--- --00	0--- --00
71Dh	PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	0000 0000	0000 0000
71Eh	PIE8	LCDIE	RTCCIE	—	—	—	SMT1PWAIE	SMT1PRAIE	SMT1IE	00-- -000	00-- -000
71Fh	—	Unimplemented								---- ----	---- ----

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Unimplemented data memory locations, read as '0'.

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19155/56/75/76/85/86 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 61 (Continued)											
1EE5h	—	Unimplemented								---- ----	---- ----
1EE6h	—	Unimplemented								---- ----	---- ----
1EE7h	—	Unimplemented								---- ----	---- ----
1EE8h	—	Unimplemented								---- ----	---- ----
1EE9h	—	Unimplemented								---- ----	---- ----
1EEAh	—	Unimplemented								---- ----	---- ----
1EEBh	—	Unimplemented								---- ----	---- ----
1EECh	—	Unimplemented								---- ----	---- ----
1EEDh	—	Unimplemented								---- ----	---- ----
1EEEh	—	Unimplemented								---- ----	---- ----
1EEFh	—	Unimplemented								---- ----	---- ----

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Unimplemented data memory locations, read as '0'.

6.0 DEVICE INFORMATION AREA

The Device Information Area (DIA) is a dedicated region in the program memory space, it is a new feature in the PIC16(L)F19155/56/75/76/85/86 family of devices. The DIA contains the calibration data for the internal temperature indicator module, stores the Microchip Unique Identifier words and the Fixed Voltage Reference voltage readings measured in mV.

The complete DIA table is shown in Table 6-1, followed by a description of each region and its functionality. The data is mapped from 8100h to 811Fh in the PIC16(L)F19155/56/75/76/85/86 family. These locations are read-only and cannot be erased or modified. The data is programmed into the device during manufacturing.

TABLE 6-1: DEVICE INFORMATION AREA

Address Range	Name of Region	Standard Device Information
8100h-8108h	MUI0	Microchip Unique Identifier (9 Words)
	MUI1	
	MUI2	
	MUI3	
	MUI4	
	MUI5	
	MUI6	
	MUI7	
	MUI8	
8109h	MUI9	1 Word Reserved
810Ah-8111h	EUI0	Unassigned (8 Words)
	EUI1	
	EUI2	
	EUI3	
	EUI4	
	EUI5	
	EUI6	
	EUI7	
8112h	TSLR1	Unassigned (1 word) Temperature indicator ADC reading at 90°C (low-range setting)
8113h	TSLR2	
8114h	TSLR3	Unassigned(1 word)
8115h	TSHR1	Unassigned (1 word) Temperature indicator ADC reading at 90°C (high-range setting)
8116h	TSHR2	
8117h	TSHR3	Unassigned (1 Word)
8118h	FVRA1X	ADC FVR1 Output voltage for 1x setting (in mV)
8119h	FVRA2X	ADC FVR1 Output Voltage for 2x setting (in mV)
811Ah	FVRA4X(1)	ADC FVR1 Output Voltage for 4x setting (in mV)
811Bh	FVRC1X	Comparator FVR2 output voltage for 1x setting (in mV)
811Ch	FVRC2X	Comparator FVR2 output voltage for 2x setting (in mV)
811Dh	FVRC4X(1)	Comparator FVR2 output voltage for 4x setting (in mV)
811Eh-811Fh		Unassigned (1 Word)

Note 1: Value not present on LF devices.

10.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Automatic Context Saving

Many peripherals produce interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 10-1.

FIGURE 10-1: INTERRUPT LOGIC

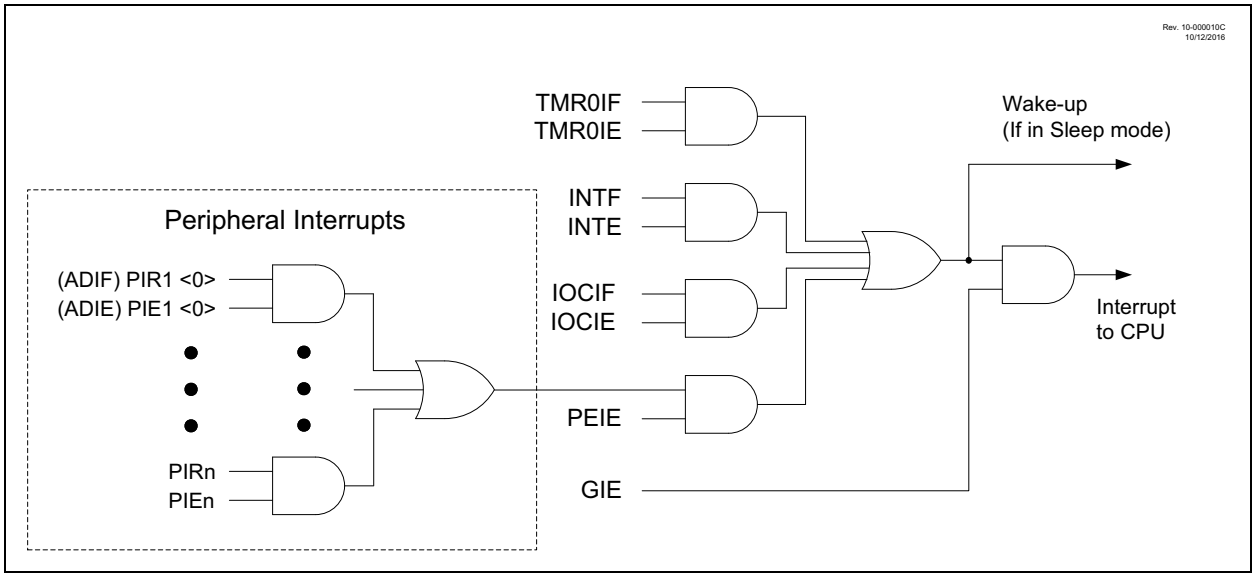
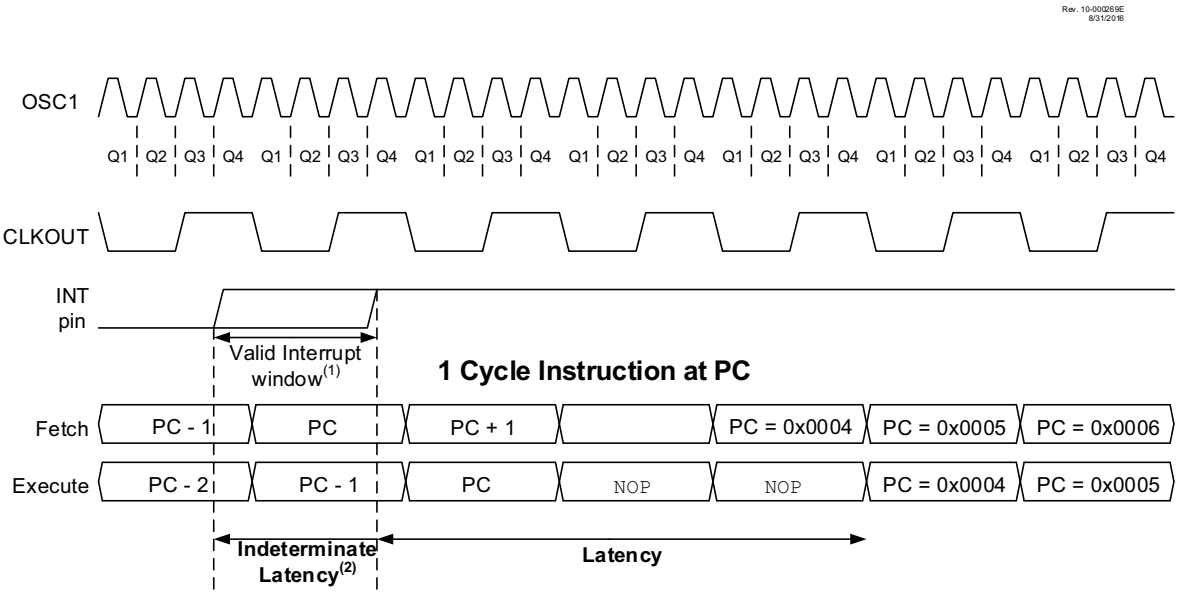
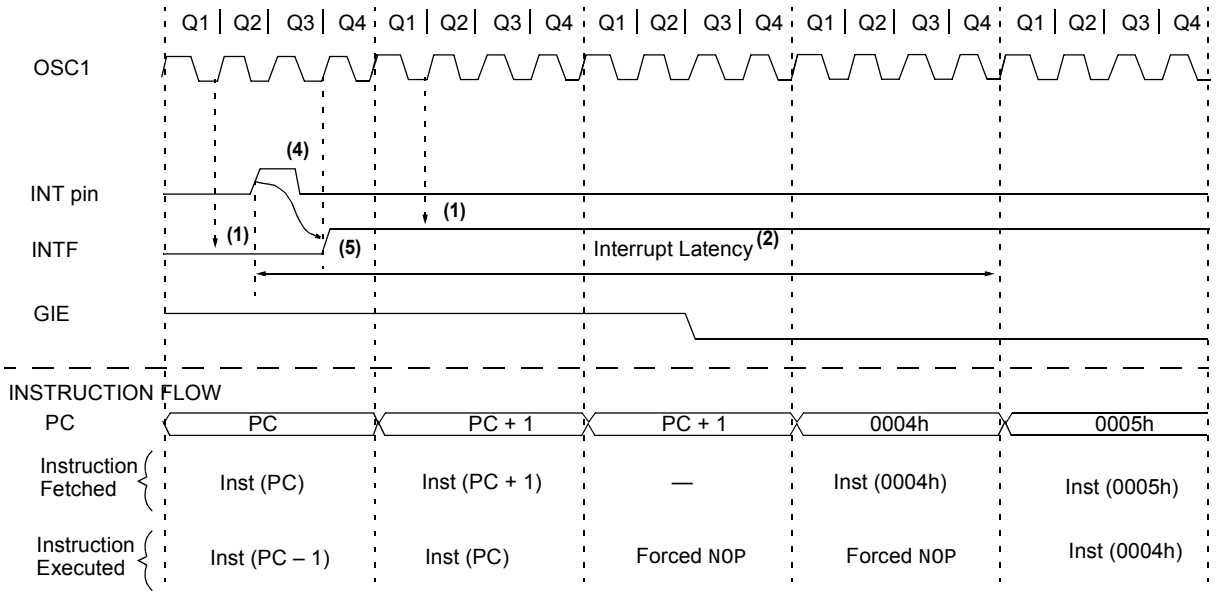


FIGURE 10-2: INTERRUPT LATENCY



Note 1: An interrupt may occur at any time during the interrupt window.
2: Since an interrupt may occur any time during the interrupt window, the actual latency can vary.

FIGURE 10-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: For minimum width of INT pulse, refer to AC specifications in **Section 39.0 "Electrical Specifications"**.

4: INTF is enabled to be set any time during the Q4-Q1 cycles.

10.3 Interrupts During Sleep

Interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to **Section 11.0 "Power-Saving Operation Modes"** for more details.

10.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. Refer to Figure 10-3. This interrupt is enabled by setting the INTE bit of the PIE0 register. The INTEDG bit of the INTCON register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the PIR0 register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

10.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{TO}$ and $\overline{PD}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

EXAMPLE 13-4: WRITING TO PROGRAM FLASH MEMORY (PFM)

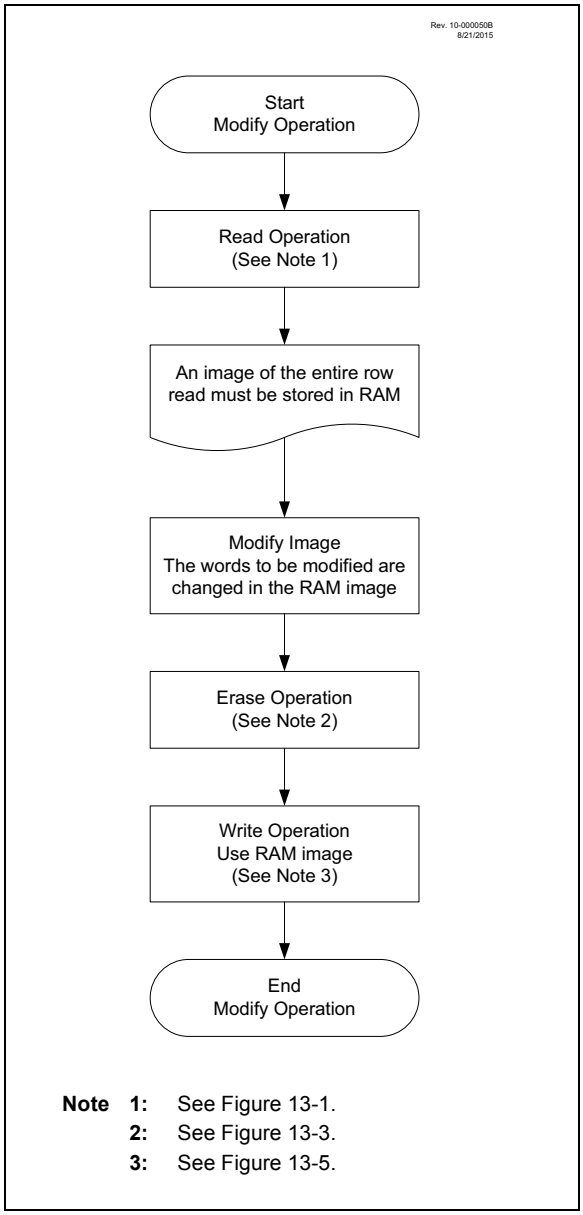
```
; This write routine assumes the following:
; 1. 32 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

        BANKSEL     NVMADRH
        MOVF        ADDRH,W
        MOVWF       NVMADRH             ; Load initial address
        MOVF        ADDRL,W
        MOVWF       NVMADRL
        MOVLW       LOW DATA_ADDR       ; Load initial data address
        MOVWF       FSR0L
        MOVLW       HIGH DATA_ADDR
        MOVWF       FSR0H
        BCF         NVMCON1,NVMREGS     ; Set Program Flash Memory as write location
        BSF         NVMCON1,WREN        ; Enable writes
        BSF         NVMCON1,LWLO        ; Load only write latches

LOOP
        MOVIW       FSR0++
        MOVWF       NVMDATL             ; Load first data byte
        MOVIW       FSR0++
        MOVWF       NVMDATH             ; Load second data byte

        MOVF        NVMADRL,W
        XORLW       0x1F                ; Check if lower bits of address are 00000
        ANDLW       0x1F                ; and if on last of 32 addresses
        BTFSC       STATUS,Z            ; Last of 32 words?
        GOTO        START_WRITE         ; If so, go write latches into memory

        CALL        UNLOCK_SEQ          ; If not, go load latch
        INCF        NVMADRL,F           ; Increment address
        GOTO        LOOP

START_WRITE
        BCF         NVMCON1,LWLO        ; Latch writes complete, now write memory
        CALL        UNLOCK_SEQ          ; Perform required unlock sequence
        BCF         NVMCON1,WREN        ; Disable writes

UNLOCK_SEQ
        MOVLW       55h
        BCF         INTCON,GIE          ; Disable interrupts
        MOVWF       NVMCON2             ; Begin unlock sequence
        MOVLW       AAh
        MOVWF       NVMCON2
        BSF         NVMCON1,WR
        BSF         INTCON,GIE          ; Unlock sequence complete, re-enable interrupts
        return
```

13.4.6 MODIFYING FLASH PROGRAM MEMORY

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 13-6: FLASH PROGRAM MEMORY MODIFY FLOWCHART



Note 1: See Figure 13-1.
2: See Figure 13-3.
3: See Figure 13-5.

27.7 Register Definitions: Timer2/4 Control

REGISTER 27-1: TxCLKCON: TIMER2/4 CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CS<3:0>			
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

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

bit 3-0 **CS<3:0>:** Timer2/4 Clock Select bits

1111 = Reserved
1110 = Reserved
1101 = Reserved
1100 = LC4_out
1011 = LC3_out
1010 = LC2_out
1001 = LC1_out
1000 = ZCD1_output
0111 = SOSC
0110 = MFINTOSC (31.25 kHz)
0101 = MFINTOSC (500 kHz)
0100 = LFINTOSC
0011 = HFINTOSC (32 MHz)
0010 = FOSC
0001 = FOSC/4
0000 = T2CKIPPS

REGISTER 29-2: CCPxCAP: CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/x	R/W-0/x	R/W-0/x
—	—	—	—	—	CTS<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **CTS<2:0>:** Capture Trigger Input Selection bits

CTS	CCP1.capture	CCP2.capture
1000	RTCC_seconds	
0111	LC4_out	
0110	LC3_out	
0101	LC2_out	
0100	LC1_out	
0011	IOC_interrupt	
0010	C2OUT	
0001	C1OUT	
0000	CCP1PPS	CCP2PPS

REGISTER 29-3: CCPRxL REGISTER: CCPx REGISTER LOW BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRx<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 CCPxMODE = Capture mode
CCPRxL<7:0>: Capture value of TMR1L
CCPxMODE = Compare mode
CCPRxL<7:0>: LS Byte compared to TMR1L
CCPxMODE = PWM modes when CCPxFMT = 0:
CCPRxL<7:0>: Pulse-width Least Significant eight bits
CCPxMODE = PWM modes when CCPxFMT = 1:
CCPRxL<7:6>: Pulse-width Least Significant two bits
CCPRxL<5:0>: Not used.

31.8 Dead-Band Uncertainty

When the rising and falling edges of the input source are asynchronous to the CWG clock, it creates uncertainty in the dead-band time delay. The maximum uncertainty is equal to one CWG clock period. Refer to Equation 31-1 for more details.

EQUATION 31-1: DEAD-BAND UNCERTAINTY

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

Example:

$FCWG\_CLOCK = 16\ MHz$

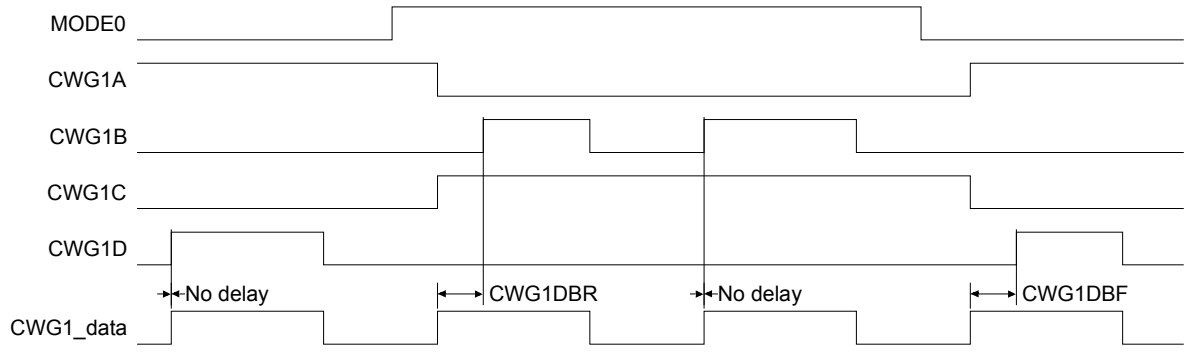
Therefore:

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

$$= \frac{1}{16MHz}$$

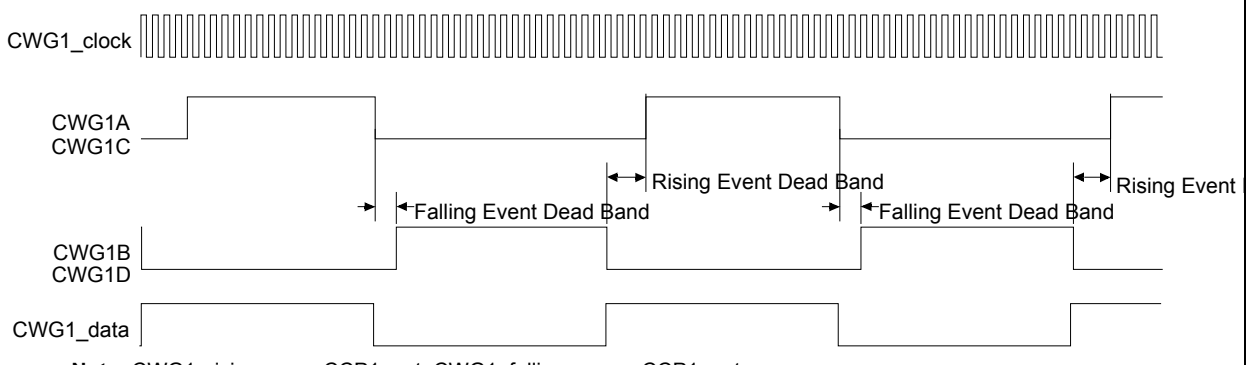
$$= 62.5ns$$

FIGURE 31-8: EXAMPLE OF PWM DIRECTION CHANGE



Note 1: WGPOL{ABCD} = 0
2: The direction bit MODE<0> (Register 31-1) can be written any time during the PWM cycle, and takes effect at the next rising CWG1_data.
3: When changing directions, CWG1A and CWG1C switch at rising CWG1_data; modulated CWG1B and CWG1D are held inactive for the dead band duration

FIGURE 31-9: CWG HALF-BRIDGE MODE OPERATION



Note: CWG1_rising_src = CCP1_out, CWG1_falling_src = ~CCP1_out

32.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

32.1.1 DATA SELECTION

There are 40 signals available as inputs to the configurable logic. Four 40-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers as indicated on the left side of Figure 32-2. Data inputs in the figure are identified by a generic numbered input name.

Table 32-2 correlates the generic input name to the actual signal for each CLC module. The column labeled 'LCxDyS<5:0> Value' indicates the MUX selection code for the selected data input. LCxDyS is an abbreviation to identify specific multiplexers: LCxD1S<5:0> through LCxD4S<5:0>.

Data inputs are selected with CLCxSEL0 through CLCxSEL3 registers (Register 32-3 through Register 32-6).

TABLE 32-2: CLCx DATA INPUT SELECTION

LCxDyS<5:0> Value	CLCx Input Source
100101 to 111111	Reserved
100100	EUSART2 (TX/CK) output
100011	EUSART2 (DT) output
100010	CWG1B output
100001	CWG1A output
100000	RTCC seconds
011111	MSSP1 SCK output
011110	MSSP1 SDO output
011101	EUSART1 (TX/CK) output
011100	EUSART1 (DT) output
011011	CLC4 output
011010	CLC3 output
011001	CLC2 output
011000	CLC1 output
010111	IOCIF
010110	ZCD output
010101	C2OUT
010100	C1OUT
010011	PWM4 output
010010	PWM3 output
010001	CCP2 output
010000	CCP1 output
001111	SMT overflow
001110	Timer4 overflow
001101	Timer2 overflow
001100	Timer1 overflow
001011	Timer0 overflow
001010	ADCRC
001001	SOSC
001000	MFINTOSC (32 kHz)
000111	MFINTOSC (500 kHz)
000110	LFINTOSC
000101	HFINTOSC
000100	FOSC
000011	CLCIN3PPS
000010	CLCIN2PPS
000001	CLCIN1PPS
000000	CLCIN0PPS

32.7 Register Definitions: CLC Control

REGISTER 32-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	—	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **LCxEN:** Configurable Logic Cell Enable bit
1 = Configurable logic cell is enabled and mixing input signals
0 = Configurable logic cell is disabled and has logic zero output

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **LCxOUT:** Configurable Logic Cell Data Output bit
Read-only: logic cell output data, after LCPOL; sampled from CLCxOUT

bit 4 **LCxINTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a rising edge occurs on CLCxOUT
0 = CLCxIF will not be set

bit 3 **LCxINTN:** Configurable Logic Cell Negative Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a falling edge occurs on CLCxOUT
0 = CLCxIF will not be set

bit 2-0 **LCxMODE<2:0>:** Configurable Logic Cell Functional Mode bits
111 = Cell is 1-input transparent latch with S and R
110 = Cell is J-K flip-flop with R
101 = Cell is 2-input D flip-flop with R
100 = Cell is 1-input D flip-flop with S and R
011 = Cell is S-R latch
010 = Cell is 4-input AND
001 = Cell is OR-XOR
000 = Cell is AND-OR

33.5.3 SLAVE TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an $\overline{ACK}$ pulse is sent by the slave on the ninth bit.

Following the $\overline{ACK}$, slave hardware clears the CKP bit and the SCL pin is held low (see **Section 33.5.6 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then the SCL pin should be released by setting the CKP bit of the SSPxCON1 register. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This $\overline{ACK}$ value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not $\overline{ACK}$), then the data transfer is complete. When the not $\overline{ACK}$ is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

33.5.3.1 Slave Mode Bus Collision

A slave receives a read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCL1IF bit of the PIR3 register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. User software can use the BCL1IF bit to handle a slave bus collision.

33.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 33-18 can be used as a reference to this list.

1. Master sends a Start condition on SDA and SCL.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit set is received by the Slave setting SSPxIF bit.
4. Slave hardware generates an $\overline{ACK}$ and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. R/$\overline{W}$ is set so CKP was automatically cleared after the $\overline{ACK}$.
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the $\overline{ACK}$ response from the master is loaded into the ACKSTAT register.
11. SSPxIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master $\overline{ACK}$s the clock will be stretched.

2: ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not $\overline{ACK}$; the clock is not held, but SSPxIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

33.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 33-33).

b) SCL is sampled low before SDA is asserted low (Figure 33-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- the Start condition is aborted,
- the BCL1IF flag is set and
- the MSSP module is reset to its Idle state (Figure 33-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 33-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note:	The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 33-33: BUS COLLISION DURING START CONDITION (SDA ONLY)

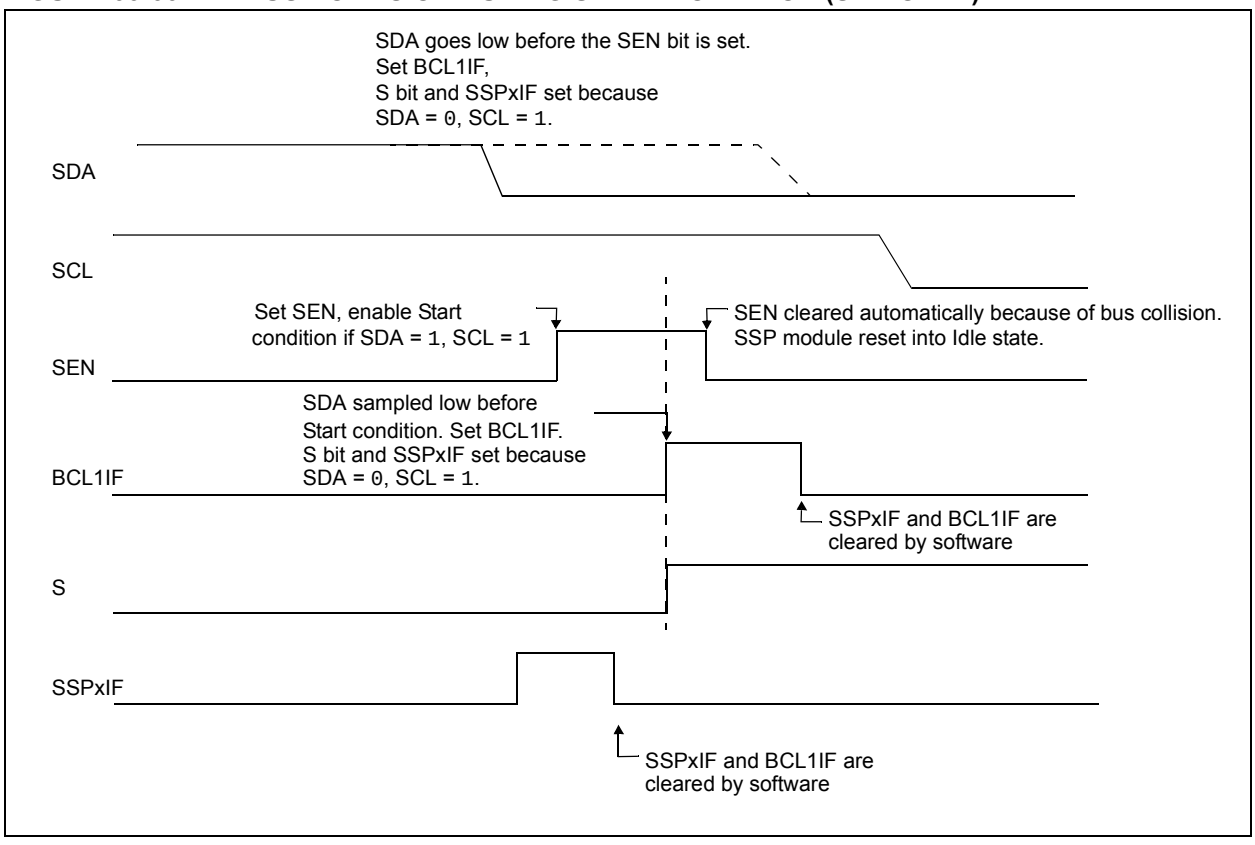
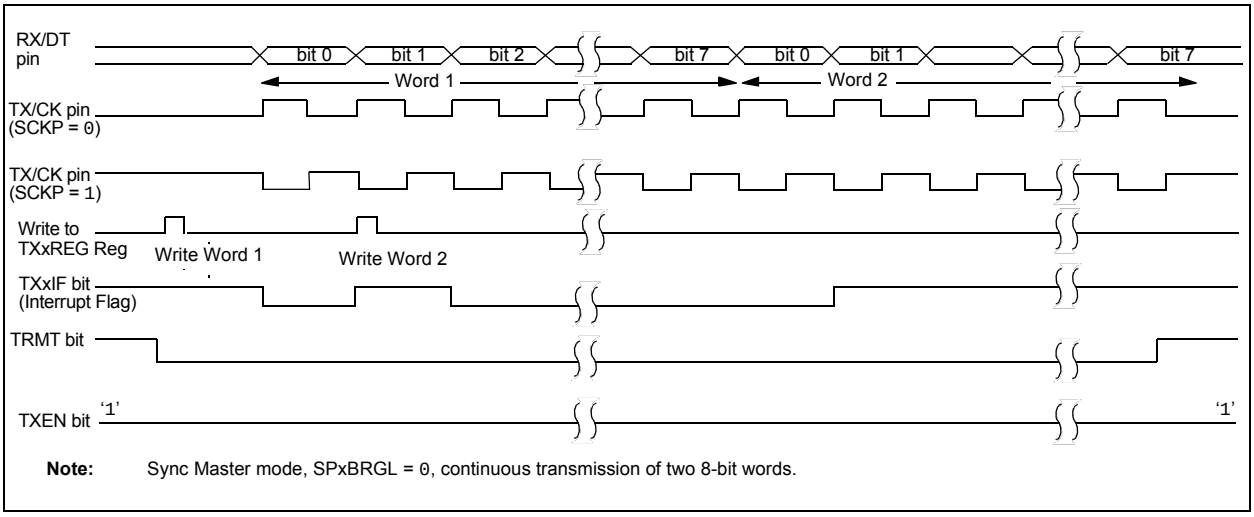
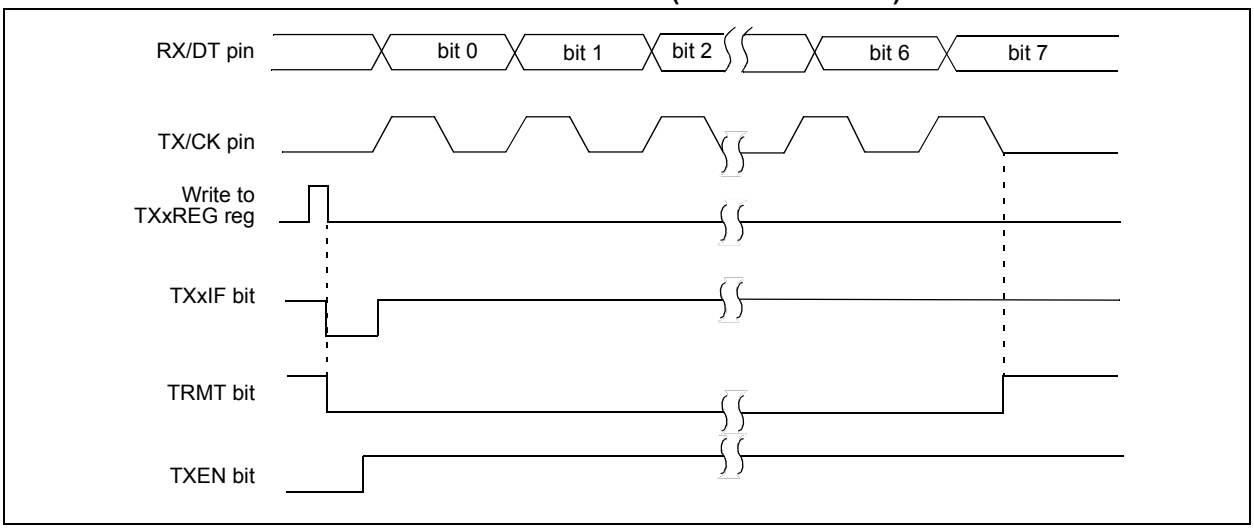


FIGURE 34-10: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SPxBRGL = 0, continuous transmission of two 8-bit words.

FIGURE 34-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



34.4.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCxSTA register) or the Continuous Receive Enable bit (CREN of the RCxSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RXxIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCxREG. The RXxIF bit remains set as long as there are unread characters in the receive FIFO.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

35.7.4 35.6.4 INTERNAL RESISTOR WITH EXTERNAL CAPACITORS

In this configuration, the user can use the internal resistor ladders to generate the LCD bias levels, and use external capacitors to guard again burst currents. It is recommend the user utilize the external capacitors when driving large glass panels with a large pixels and a high pixel count. The external capacitors will help dampen current spikes during segment switching. Contrast is adjusted using the LCDCST<2:0> bits. The CFLYx pins are available as a GPIO. See Figure 35-7 for supported connections.

External capacitors can be used when voltage to the internal resistor ladder is supplied by VDD (LCDVSRC<3:0> = 0101) or an external source (LCDVSRC<3:0> = 0100). When supplying an external voltage to internal resistor ladder the external capacitors should be limited to VLCD2, and VLCD3.

FIGURE 35-9: TYPE-A WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE

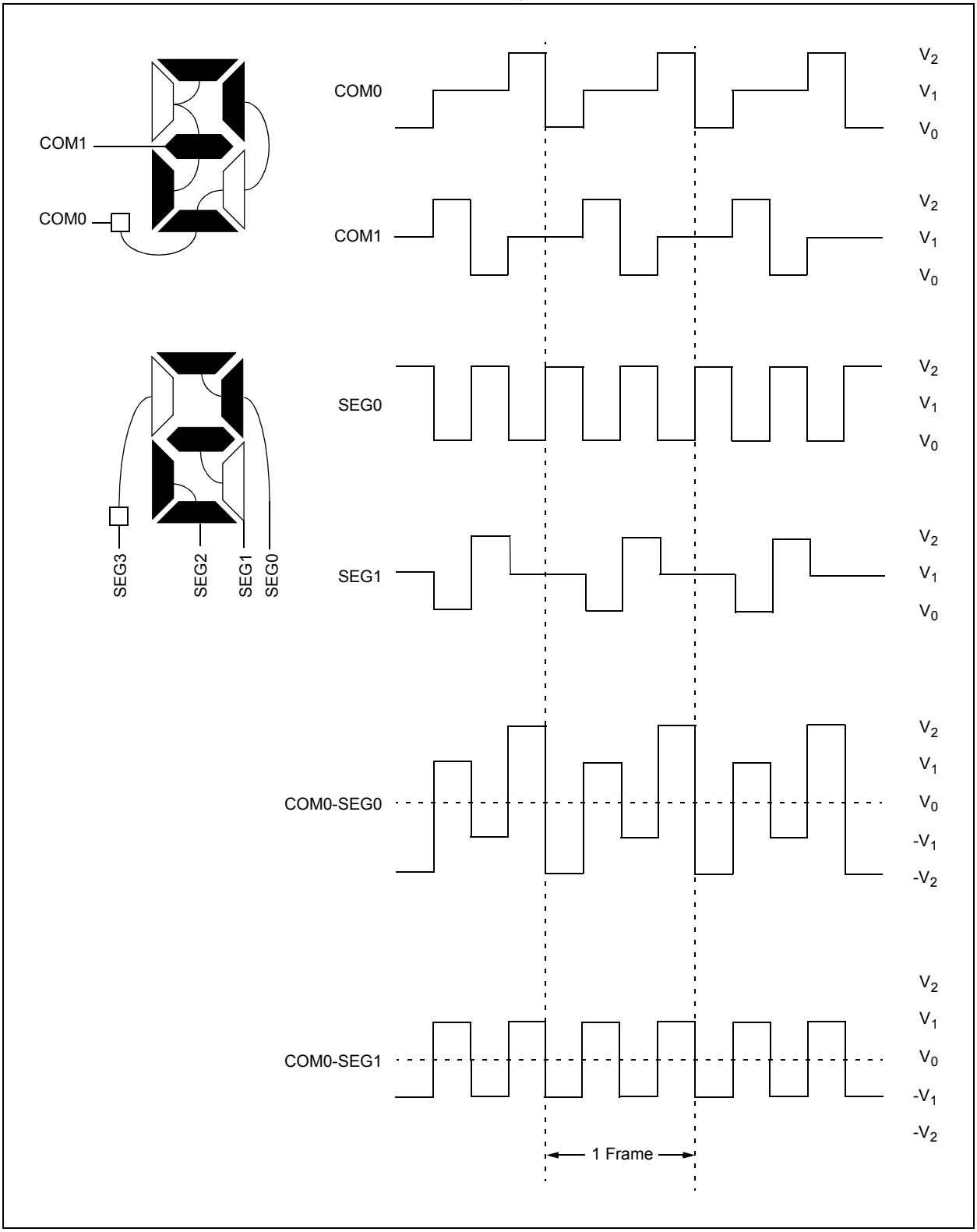


TABLE 39-26: I^2C BUS DATA REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Symbol**	**Characteristic**		**Min.**	**Max.**	**Units**	**Conditions**
SP100*	THIGH	Clock high time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP module	1.5TCY	—		
SP101*	TLOW	Clock low time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP module	1.5TCY	—		
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	
			400 kHz mode	20 + 0.1CB	250	ns	CB is specified to be from 10-400 pF
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
SP109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
SP111	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I^2C bus device can be used in a Standard mode (100 kHz) I^2C bus system, but the requirement $T_{SU:DAT} \geq 250$ ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line T_R max. + $T_{SU:DAT}$ = 1000 + 250 = 1250 ns (according to the Standard mode I^2C bus specification), before the SCL line is released.