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	24
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 20x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19156-e-ss

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 62 (Continued)											
1F20h	RC0PPS	—	—	—	RC0PPS4	RC0PPS3	RC0PPS2	RC0PPS1	RC0PPS0	0000 0000	--uu uuuu
1F21h	RC1PPS	—	—	—	RC1PPS4	RC1PPS3	RC1PPS2	RC1PPS1	RC1PPS0	0000 0000	--uu uuuu
1F22h	RC2PPS	—	—	—	RC2PPS4	RC2PPS3	RC2PPS2	RC2PPS1	RC2PPS0	0000 0000	--uu uuuu
1F23h	RC3PPS	—	—	—	RC3PPS4	RC3PPS3	RC3PPS2	RC3PPS1	RC3PPS0	0000 0000	--uu uuuu
1F24h	RC4PPS	—	—	—	RC4PPS4	RC4PPS3	RC4PPS2	RC4PPS1	RC4PPS0	0000 0000	--uu uuuu
1F25h	—	—	—	—	—	—	—	—	—	---- ----	---- ----
1F26h	RC6PPS	—	—	—	RC6PPS4	RC6PPS3	RC6PPS2	RC6PPS1	RC6PPS0	0000 0000	--uu uuuu
1F27h	RC7PPS	—	—	—	RC7PPS4	RC7PPS3	RC7PPS2	RC7PPS1	RC7PPS0	0000 0000	--uu uuuu
1F28h	RD0PPS	—	—	—	RD0PPS4	RD0PPS3	RD0PPS2	RD0PPS1	RD0PPS0	0000 0000	--uu uuuu
1F29h	RD1PPS	—	—	—	RD1PPS4	RD1PPS3	RD1PPS2	RD1PPS1	RD1PPS0	0000 0000	--uu uuuu
1F2Ah	RD2PPS	—	—	—	RD2PPS4	RD2PPS3	RD2PPS2	RD2PPS1	RD2PPS0	0000 0000	--uu uuuu
1F2Bh	RD3PPS	—	—	—	RD3PPS4	RD3PPS3	RD3PPS2	RD3PPS1	RD3PPS0	0000 0000	--uu uuuu
1F2Ch	RD4PPS	—	—	—	RD4PPS4	RD4PPS3	RD4PPS2	RD4PPS1	RD4PPS0	0000 0000	--uu uuuu
1F2Dh	RD5PPS	—	—	—	RD5PPS4	RD5PPS3	RD5PPS2	RD5PPS1	RD5PPS0	0000 0000	--uu uuuu
1F2Eh	RD6PPS	—	—	—	RD6PPS4	RD6PPS3	RD6PPS2	RD6PPS1	RD6PPS0	0000 0000	--uu uuuu
1F2Fh	RD7PPS	—	—	—	RD7PPS4	RD7PPS3	RD7PPS2	RD7PPS1	RD7PPS0	0000 0000	--uu uuuu
1F30h	RE0PPS	—	—	—	RE0PPS4	RE0PPS3	RE0PPS2	RE0PPS1	RE0PPS0	0000 0000	--uu uuuu
1F31h	RE1PPS	—	—	—	RE1PPS4	RE1PPS3	RE1PPS2	RE1PPS1	RE1PPS0	0000 0000	--uu uuuu
1F32h	RE2PPS	—	—	—	RE2PPS4	RE2PPS3	RE2PPS2	RE2PPS1	RE2PPS0	0000 0000	--uu uuuu
1F33h	—	Unimplemented								---- ----	---- ----
1F34h	—	Unimplemented								---- ----	---- ----
1F35h	—	Unimplemented								---- ----	---- ----
1F36h	—	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'.

4.6.2 LINEAR DATA MEMORY

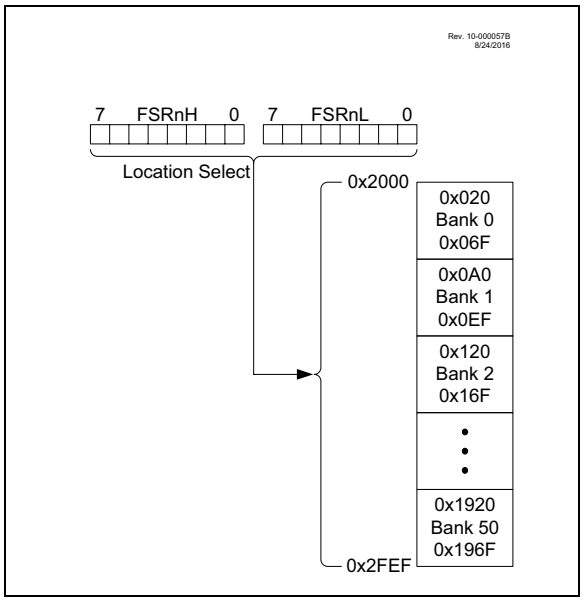
The linear data memory is the region from FSR address 0x2000 to FSR address 0X2FEF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks. Refer to Figure 4-8 and Figure 4-9 for the Linear Data Memory Map.

Note:	The address range 0x2000 to 0x2FF0 represents the complete addressable Linear Data Memory up to Bank 50. The actual implemented Linear Data Memory will differ from one device to the other in a family. Confirm the memory limits on every device.

Unimplemented memory reads as `0x00`. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

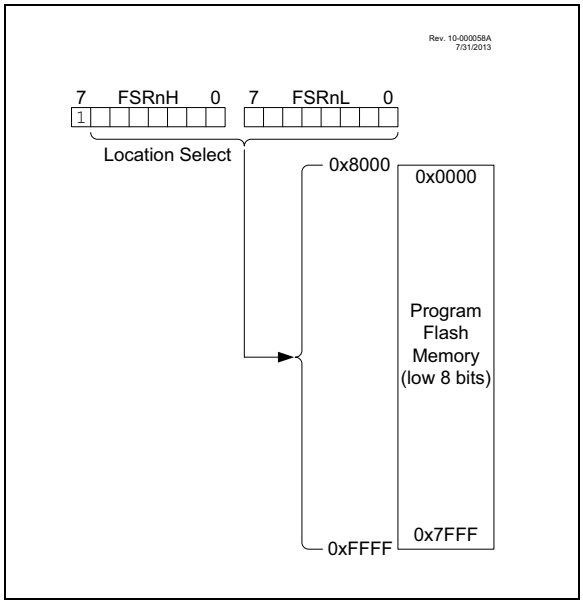
FIGURE 4-11: LINEAR DATA MEMORY MAP



4.6.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire Program Flash Memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the Program Flash Memory cannot be accomplished via the FSR/INDF interface. All instructions that access Program Flash Memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 4-12: PROGRAM FLASH MEMORY MAP



4.6.4 DATA EEPROM MEMORY

The EEPROM memory can be read or written through the NVMCON register interface (see **Section 13.2 "Data EEPROM Memory"**). However, to make access to the EEPROM easier, read-only access to the EEPROM contents are also available through indirect addressing via an FSR. When the MSP of the FSR (ex: FSRxH) is set to 0x70, the lower 8-bit address value (in FSRxL) determines the EEPROM location that may be read (via the INDF register). In other words, the EEPROM address range 0x00-0xFF is mapped into the FSR address space between 0x7000 and 0x70FF. Writing to the EEPROM cannot be accomplished via the FSR/INDF interface. Reads from the EEPROM through the FSR/INDF interface will require one additional instruction cycle to complete.

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

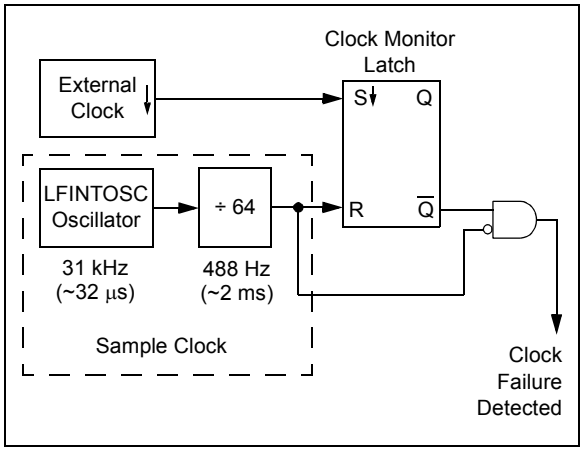
Address Range	Name of Region	Standard Device Information
8100h-8108h	MUI0	Microchip Unique Identifier (9 Words)
	MUI1	
	MUI2	
	MUI3	
	MUI4	
	MUI5	
	MUI6	
	MUI7	
	MUI8	
8109h	MUI9	1 Word Reserved
810Ah-8111h	EUI0	Unassigned (8 Words)
	EUI1	
	EUI2	
	EUI3	
	EUI4	
	EUI5	
	EUI6	
	EUI7	
8112h	TSLR1	Unassigned (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.

9.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to all external Oscillator modes (ECL, ECM, ECH and Secondary Oscillator).

FIGURE 9-7: FSCM BLOCK DIAGRAM



9.4.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64 (see Figure 9-7). Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

9.4.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to the HFINTOSC at 1 MHz clock frequency and sets the bit flag OSFIF of the PIR1 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE1 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation, by writing to the NOSC and NDIV bits of the OSCCON1 register.

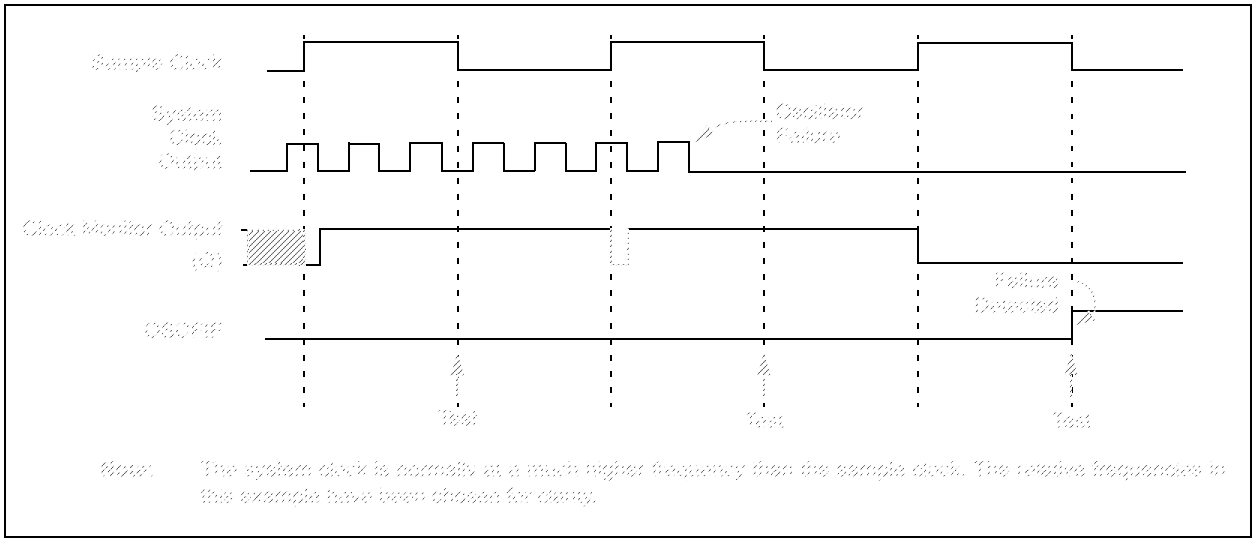
9.4.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or changing the NOSC and NDIV bits of the OSCCON1 register. When switching to the external oscillator, or external oscillator and PLL, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON1. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSFIF flag will again become set by hardware.

9.4.4 RESET OR WAKE-UP FROM SLEEP

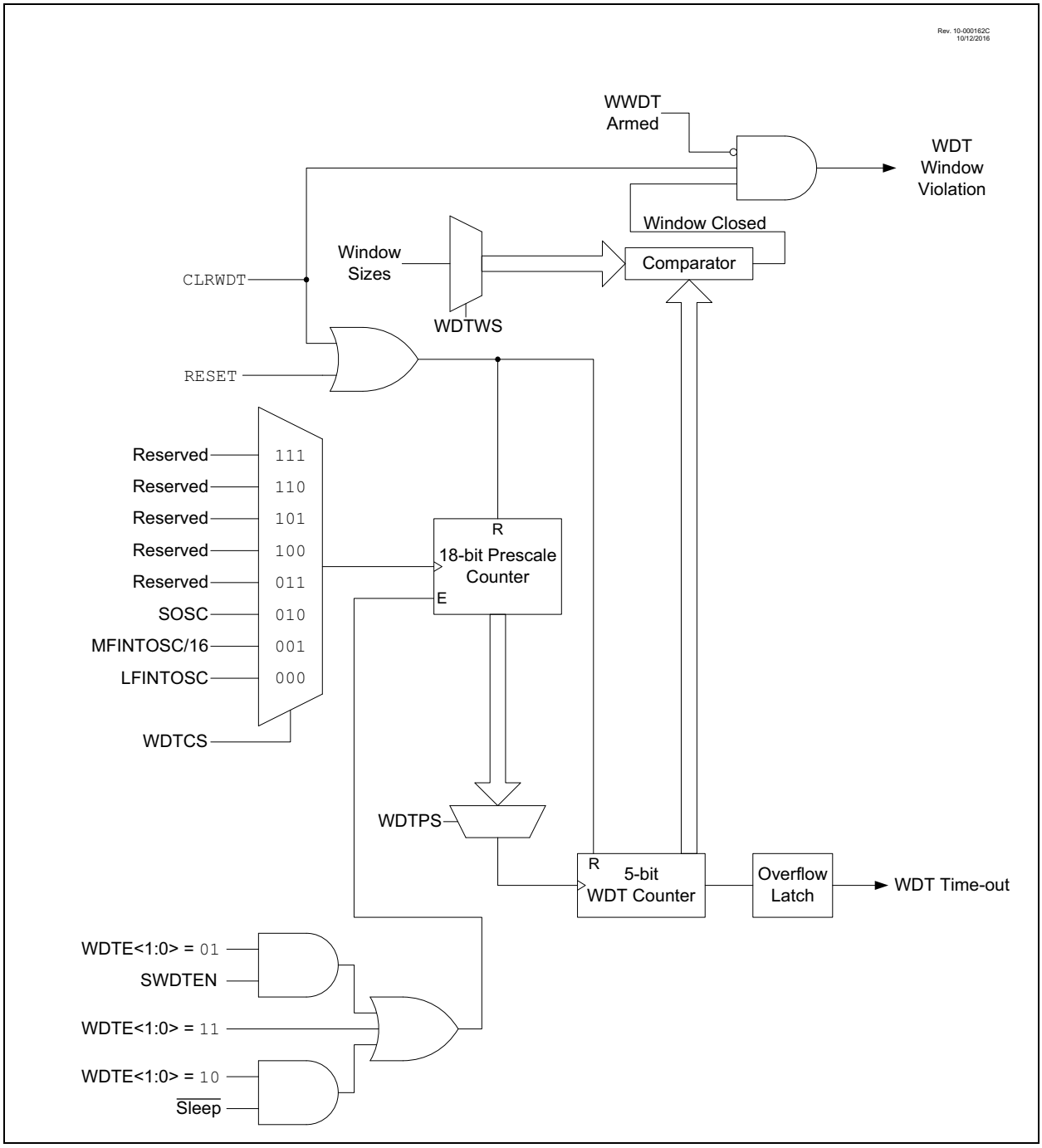
The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. Therefore, the device will always be executing code while the OST is operating.

FIGURE 9-8: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

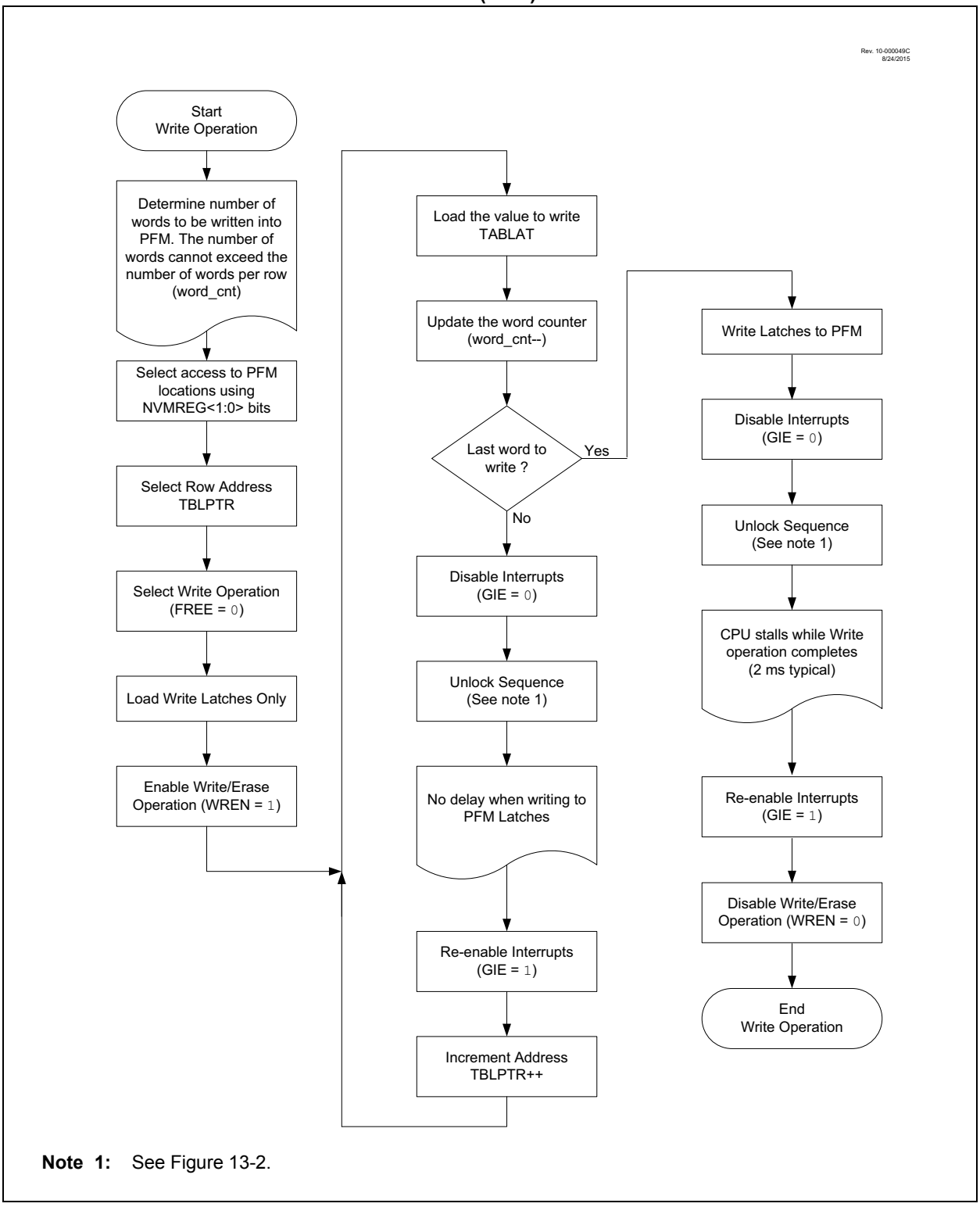
FIGURE 12-1: WATCHDOG TIMER BLOCK DIAGRAM



EXAMPLE 13-3: ERASING ONE ROW OF PROGRAM FLASH MEMORY (PFM)

```
; This sample row erase routine assumes the following:
; 1.A valid address within the erase row is loaded in variables ADDRH:ADDRL
; 2.ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)

BANKSEL         NVMADRL
MOVF            ADDRL,W
MOVWF           NVMADRL                 ; Load lower 8 bits of erase address boundary
MOVF            ADDRH,W
MOVWF           NVMADRH                 ; Load upper 6 bits of erase address boundary
BCF             NVMCON1,NVMREGS         ; Choose PFM memory area
BSF             NVMCON1,FREE            ; Specify an erase operation
BSF             NVMCON1,WREN            ; Enable writes
BCF             INTCON,GIE              ; Disable interrupts during unlock sequence

; ------------------------------REQUIRED UNLOCK SEQUENCE:------------------------------

MOVLW           55h                     ; Load 55h to get ready for unlock sequence
MOVWF           NVMCON2                 ; First step is to load 55h into NVMCON2
MOVLW           AAh                     ; Second step is to load AAh into W
MOVWF           NVMCON2                 ; Third step is to load AAh into NVMCON2
BSF             NVMCON1,WR              ; Final step is to set WR bit

; -------------------------------------------------------------------------------------

BSF             INTCON,GIE              ; Re-enable interrupts, erase is complete
BCF             NVMCON1,WREN            ; Disable writes
```

FIGURE 13-5: PROGRAM FLASH MEMORY (PFM) WRITE FLOWCHART



Note 1: See Figure 13-2.

REGISTER 14-11: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
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 7-0 **LATB<7:0>**: RB<7:0> Output Latch Value bits(1)

Note 1: Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register is return of actual I/O pin values.

REGISTER 14-12: ANSELB: PORTB ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
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 7-0 **ANSB<7:0>**: Analog Select between Analog or Digital Function on pins RB<7:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

FIGURE 24-2: TIMER DIGIT FORMAT

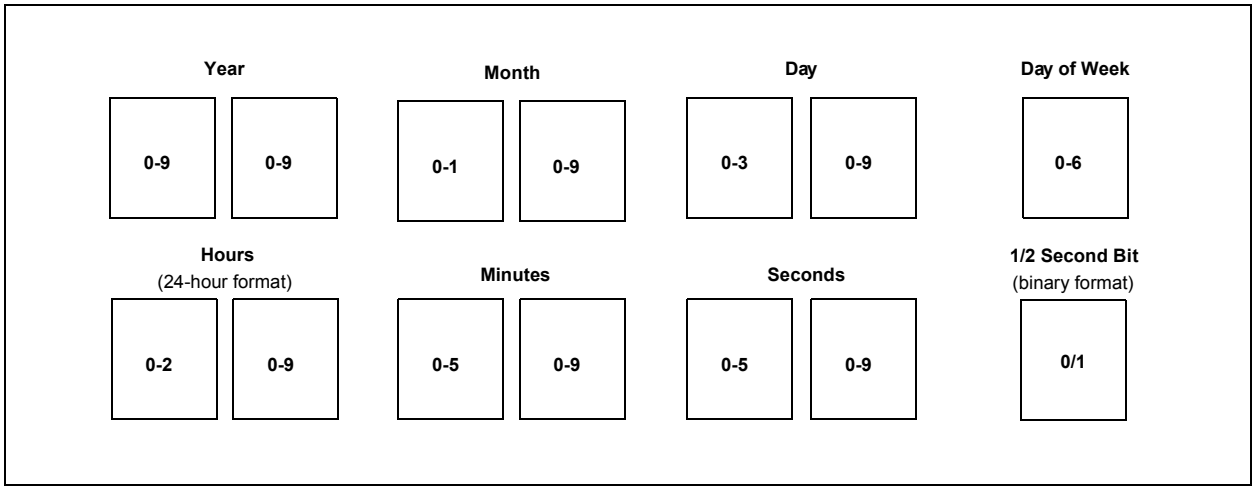


FIGURE 24-3: ALARM DIGIT FORMAT

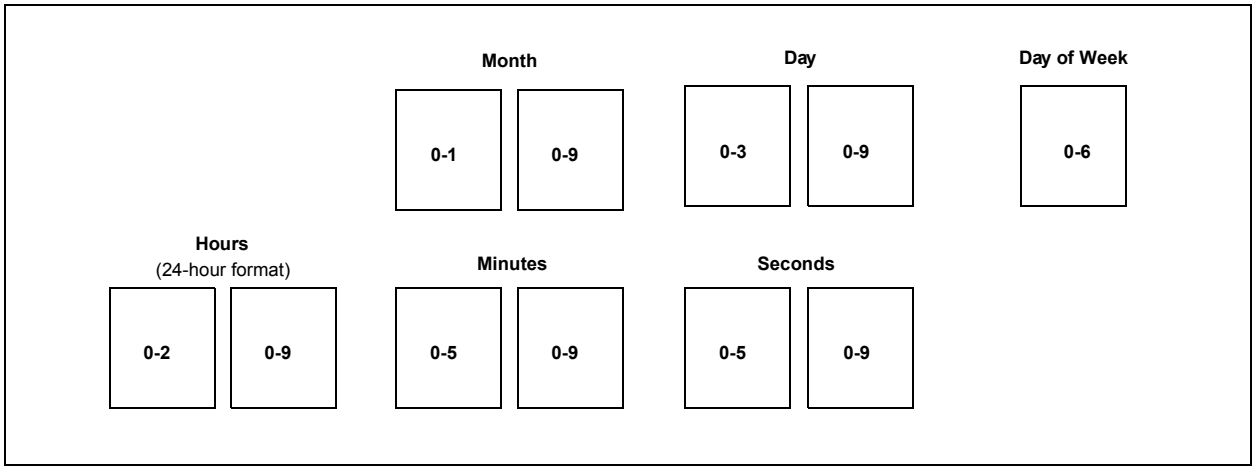
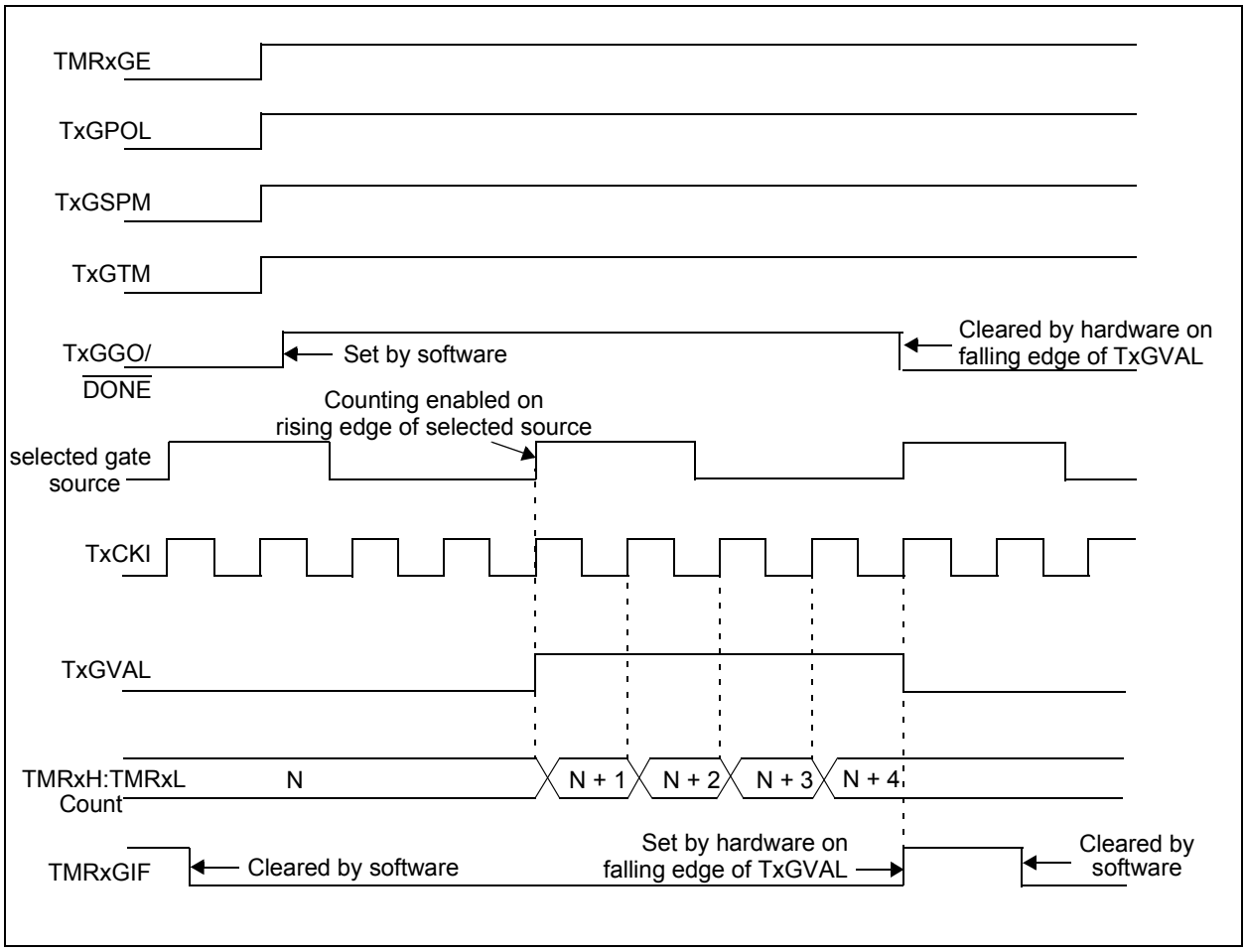


FIGURE 26-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



26.11 Peripheral Module Disable

When a peripheral module is not used or inactive, the module can be disabled by setting the Module Disable bit in the PMD registers. This will reduce power consumption to an absolute minimum. Setting the PMD bits holds the module in Reset and disconnects the module's clock source. The Module Disable bit for Timer1 (TMR1MD) are in the PMD1 register. See **Section 16.0 "Peripheral Module Disable (PMD)"** for more information.

28.7.5 WINDOWED MEASURE MODE

Windowed Measure mode measures the window duration of the SMTWINx input of the SMT. It begins incrementing the timer on a rising edge of the SMTWINx input and updates the SMTxCPR register with the value of the timer and resets the timer on a second rising edge. See Figure 28-10 and Figure 28-11.

31.9 CWG Steering Mode

In Steering mode (MODE = 00x), the CWG allows any combination of the CWG1x pins to be the modulated signal. The same signal can be simultaneously available on multiple pins, or a fixed-value output can be presented.

When the respective STRx bit of CWG1OCON0 is '0', the corresponding pin is held at the level defined. When the respective STRx bit of CWG1OCON0 is '1', the pin is driven by the input data signal. The user can assign the input data signal to one, two, three, or all four output pins.

The POLx bits of the CWG1CON1 register control the signal polarity only when STRx = 1.

The CWG auto-shutdown operation also applies in Steering modes as described in **Section 31.10 "Auto-Shutdown"**. An auto-shutdown event will only affect pins that have STRx = 1.

31.9.1 STEERING SYNCHRONIZATION

Changing the MODE bits allows for two modes of steering, synchronous and asynchronous.

When MODE = 000, the steering event is asynchronous and will happen at the end of the instruction that writes to STRx (that is, immediately). In this case, the output signal at the output pin may be an incomplete waveform. This can be useful for immediately removing a signal from the pin.

When MODE = 001, the steering update is synchronous and occurs at the beginning of the next rising edge of the input data signal. In this case, steering the output on/off will always produce a complete waveform.

Figure 31-10 and Figure 31-11 illustrate the timing of asynchronous and synchronous steering, respectively.

FIGURE 31-10: EXAMPLE OF ASYNCHRONOUS STEERING EVENT (MODE<2:0> = 000)

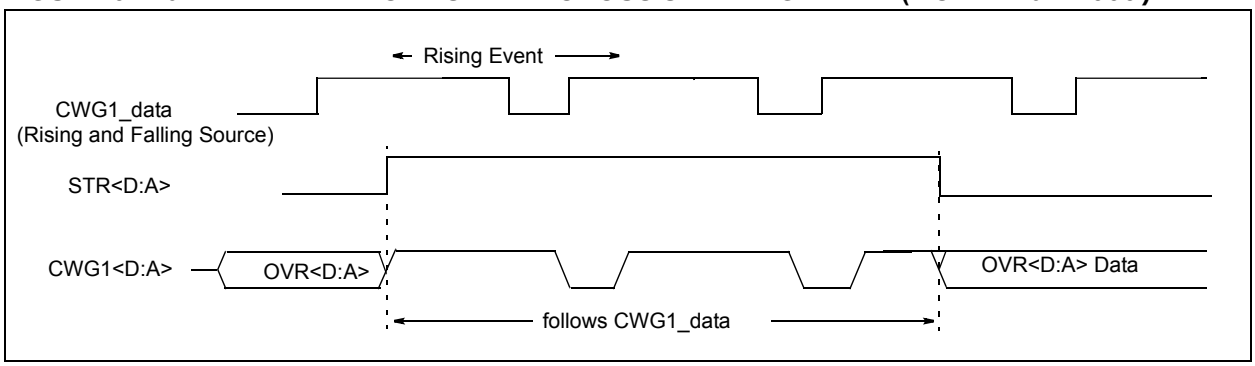
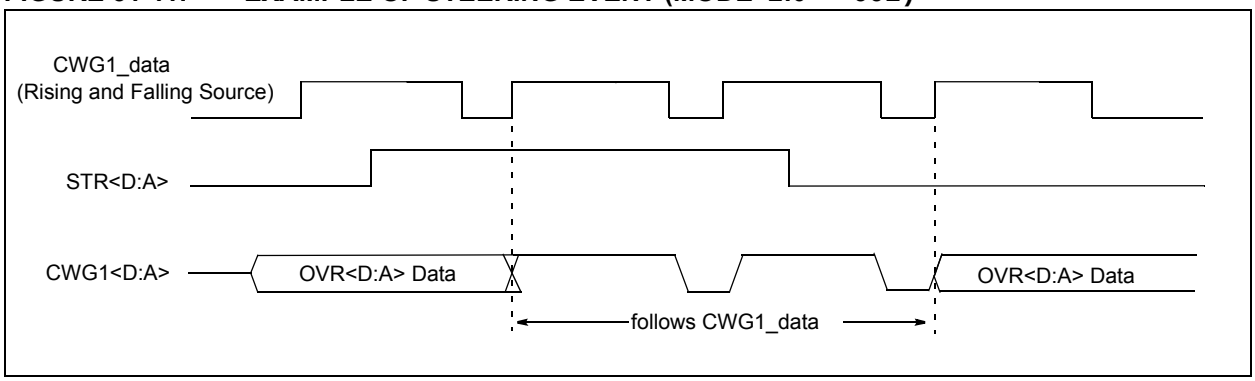


FIGURE 31-11: EXAMPLE OF STEERING EVENT (MODE<2:0> = 001)



33.5 I²C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected by the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

33.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 33-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 33-5) affects the address matching process. See **Section 33.5.9 "SSP Mask Register"** for more information.

33.5.1.1 I²C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

33.5.1.2 I²C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

33.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPxSTAT register is set, or bit SSPOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 33-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSPxIF, must be cleared by software.

When the SEN bit of the SSPxCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPxCON1 register.

33.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I²C slave in 7-bit Addressing mode. Figure 33-14 and Figure 33-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I²C communication.

1. Start bit detected.
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 clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes idle.

REGISTER 34-4: RCxREG(1): RECEIVE DATA REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RCxREG<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 Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **RCxREG<7:0>:** Lower eight bits of the received data; read-only; see also RX9D (Register 34-2)

Note 1: RCxREG (including the 9th bit) is double buffered, and data is available while new data is being received.

REGISTER 34-5: TXxREG(1): TRANSMIT DATA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXxREG<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 Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **TXxREG<7:0>:** Lower eight bits of the received data; read-only; see also RX9D (Register 34-1)

Note 1: TXxREG (including the 9th bit) is double buffered, and can be written when previous data has started shifting.

REGISTER 34-6: SPxBRGL(1): BAUD RATE GENERATOR REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPxBRG<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 Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **SPxBRG<7:0>:** Lower eight bits of the Baud Rate Generator

Note 1: Writing to SP1BRG resets the BRG counter.

TABLE 35-5: LCDDATAX REGISTERS AND BITS FOR SEGMENT AND COM COMBINATIONS (48-PIN)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDDATA0	SEG7 COM0	SEG6 COM0	—	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	—	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
LCDDATA2	SEG23 COM0	SEG22 COM0	—	SEG20 COM0	SEG19 COM0	SEG18 COM0	—	—
LCDDATA3	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCDDATA4	—	—	—	—	—	SEG34 COM0	SEG33 COM0	SEG32 COM0
LCDDATA5	SEG47 COM0	SEG46 COM0	SEG45 COM0	SEG44 COM0	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0
LCDDATA6	SEG7 COM1	SEG6 COM1	—	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
LCDDATA7	SEG15 COM1	SEG14 COM1	SEG13 COM1	—	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
LCDDATA8	SEG23 COM1	SEG22 COM1	—	SEG20 COM1	SEG19 COM1	SEG18 COM1	—	—
LCDDATA9	SEG31 COM1	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCDDATA10	—	—	—	—	—	SEG34 COM1	SEG33 COM1	SEG32 COM1
LCDDATA11	SEG47 COM1	SEG46 COM1	SEG45 COM1	SEG44 COM1	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1
LCDDATA12	SEG7 COM2	SEG6 COM2	—	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
LCDDATA13	SEG15 COM2	SEG14 COM2	SEG13 COM2	—	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
LCDDATA14	SEG23 COM2	SEG22 COM2	—	SEG20 COM2	SEG19 COM2	SEG18 COM2	—	—
LCDDATA15	SEG31 COM2	SEG30 COM2	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCDDATA16	—	—	—	—	—	SEG34 COM2	SEG33 COM2	SEG32 COM2
LCDDATA17	SEG47 COM2	SEG46 COM2	SEG45 COM2	SEG44 COM2	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2
LCDDATA18	SEG7 COM3	SEG6 COM3	—	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
LCDDATA19	SEG15 COM3	SEG14 COM3	SEG13 COM3	—	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
LCDDATA20	SEG23 COM3	SEG22 COM3	—	SEG20 COM3	SEG19 COM3	SEG18 COM3	—	—
LCDDATA21	SEG31 COM3	SEG30 COM3	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCDDATA22	—	—	—	—	—	SEG34 COM3	SEG33 COM3	SEG32 COM3
LCDDATA23	SEG47 COM3	SEG46 COM3	SEG45 COM3	SEG44 COM3	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3
LCDDATA24	SEG7 COM4	SEG6 COM4	—	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
LCDDATA25	SEG15 COM4	SEG14 COM4	SEG13 COM4	—	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
LCDDATA26	SEG23 COM4	SEG22 COM4	—	SEG20 COM4	SEG19 COM4	SEG18 COM4	—	—
LCDDATA27	SEG31 COM4	SEG30 COM4	SEG29 COM4	SEG28 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCDDATA28	—	—	—	—	—	SEG34 COM4	SEG33 COM4	SEG32 COM4
LCDDATA29	SEG47 COM4	SEG46 COM4	SEG45 COM4	SEG44 COM4	SEG43 COM4	SEG42 COM4	SEG41 COM4	SEG40 COM4
LCDDATA30	SEG7 COM5	SEG6 COM5	—	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
LCDDATA31	SEG15 COM5	SEG14 COM5	SEG13 COM5	—	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
LCDDATA32	SEG23 COM5	SEG22 COM5	—	SEG20 COM5	SEG19 COM5	SEG18 COM5	—	—
LCDDATA33	SEG31 COM5	SEG30 COM5	SEG29 COM5	SEG28 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCDDATA34	—	—	—	—	—	SEG34 COM5	SEG33 COM5	SEG32 COM5
LCDDATA35	SEG47 COM5	SEG46 COM5	SEG45 COM5	SEG44 COM5	SEG43 COM5	SEG42 COM5	SEG41 COM5	SEG40 COM5
LCDDATA36	SEG7 COM6	SEG6 COM6	—	SEG4 COM6	SEG3 COM6	SEG2 COM6	SEG1 COM6	SEG0 COM6
LCDDATA37	SEG15 COM6	SEG14 COM6	SEG13 COM6	—	SEG11 COM6	SEG10 COM6	SEG9 COM6	SEG8 COM6
LCDDATA38	SEG23 COM6	SEG22 COM6	—	SEG20 COM6	SEG19 COM6	SEG18 COM6	—	—
LCDDATA39	SEG31 COM6	SEG30 COM6	SEG29 COM6	SEG28 COM6	SEG27 COM6	SEG26 COM6	SEG25 COM6	SEG24 COM6
LCDDATA40	—	—	—	—	—	SEG34 COM6	SEG33 COM6	SEG32 COM6
LCDDATA41	SEG47 COM6	SEG46 COM6	SEG45 COM6	SEG44 COM6	SEG43 COM6	SEG42 COM6	SEG41 COM6	SEG40 COM6
LCDDATA42	SEG7 COM7	SEG6 COM7	—	SEG4 COM7	SEG3 COM7	SEG2 COM7	SEG1 COM7	SEG0 COM7
LCDDATA43	SEG15 COM7	SEG14 COM7	SEG13 COM7	—	SEG11 COM7	SEG10 COM7	SEG9 COM7	SEG8 COM7
LCDDATA44	SEG23 COM7	SEG22 COM7	—	SEG20 COM7	SEG19 COM7	SEG18 COM7	—	—
LCDDATA45	SEG31 COM7	SEG30 COM7	SEG29 COM7	SEG28 COM7	SEG27 COM7	SEG26 COM7	SEG25 COM7	SEG24 COM7
LCDDATA46	—	—	—	—	—	SEG34 COM7	SEG33 COM7	SEG32 COM7
LCDDATA47	SEG47 COM7	SEG46 COM7	SEG45 COM7	SEG44 COM7	SEG43 COM7	SEG42 COM7	SEG41 COM7	SEG40 COM7

FIGURE 35-8: TYPE-A/TYPE-B WAVEFORMS IN STATIC DRIVE

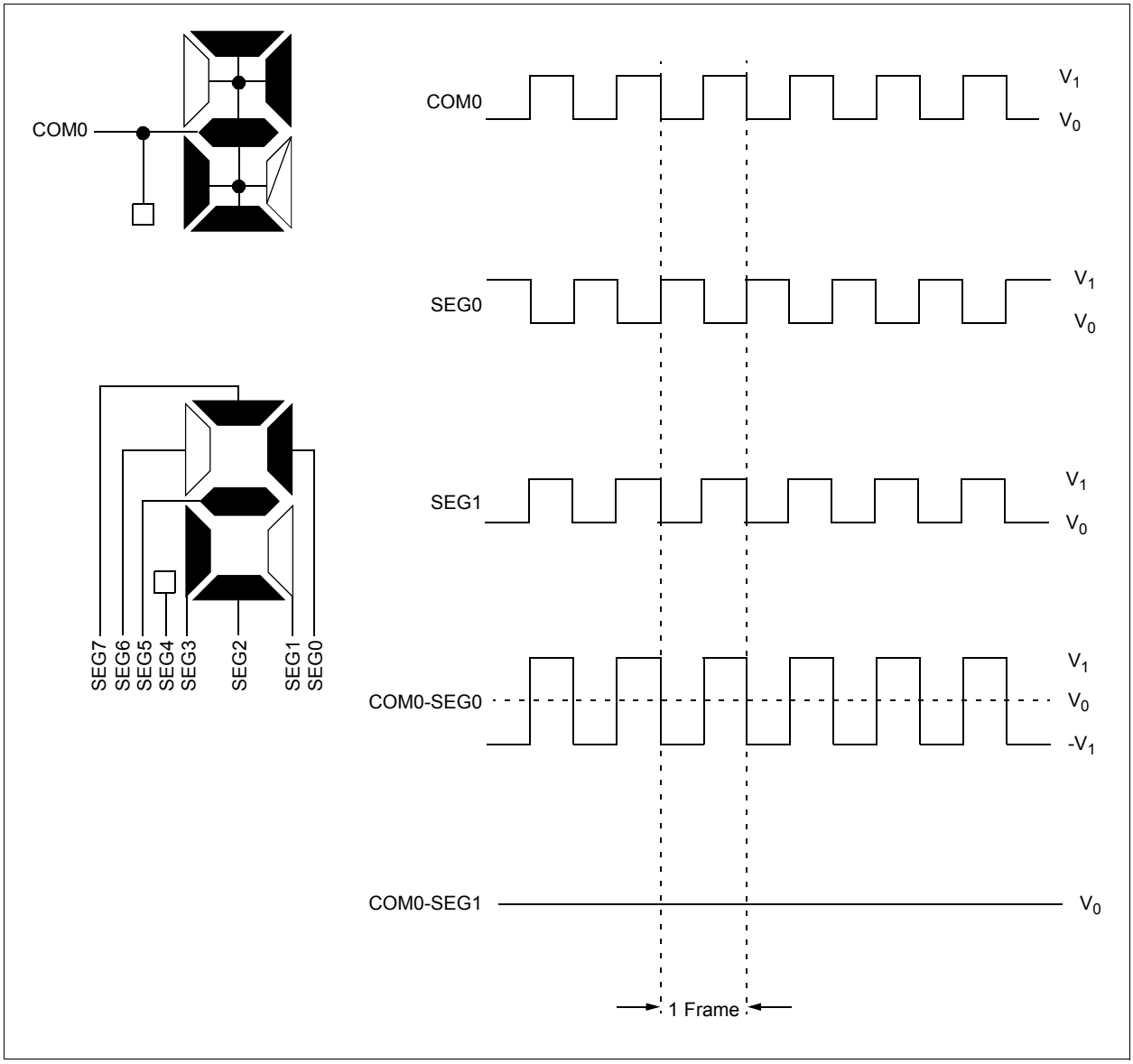


FIGURE 35-13: TYPE-A WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE

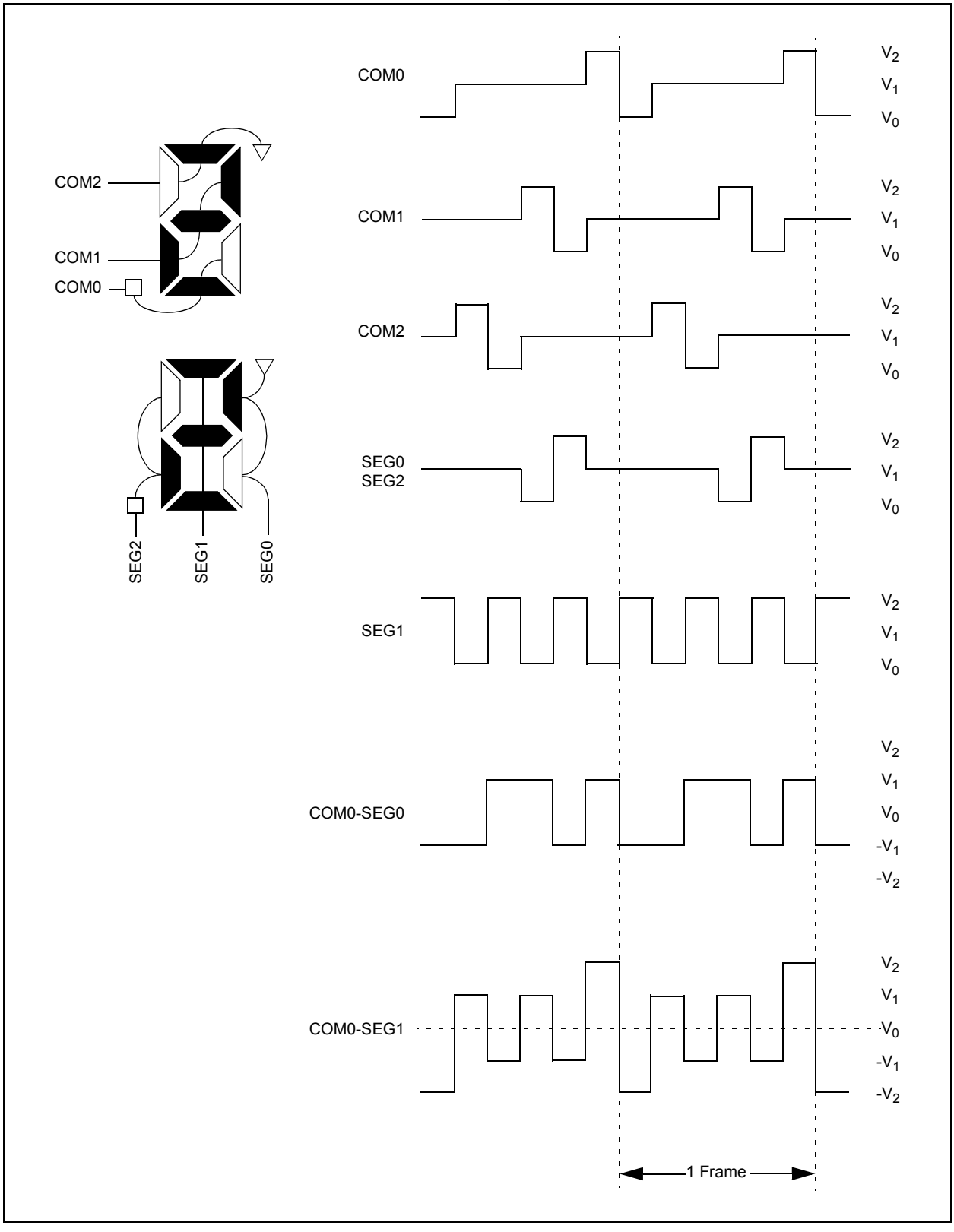
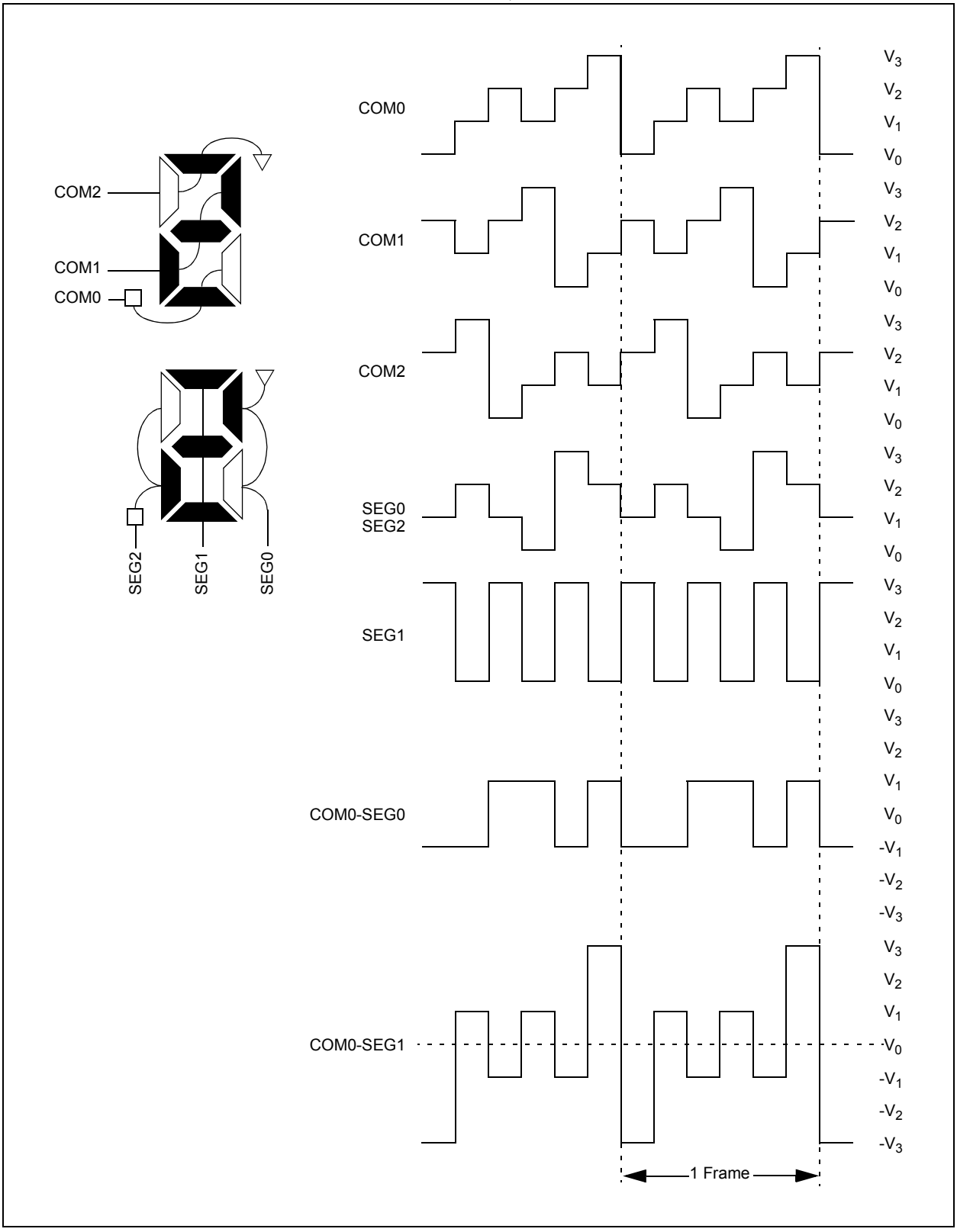


FIGURE 35-15: TYPE-A WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE



38.0 REGISTER SUMMARY

TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES

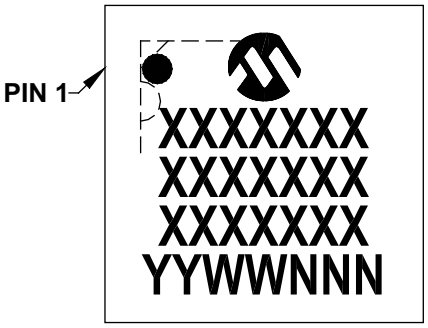
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
0Ch	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	222
0Dh	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	229
0Eh	PORTC	RC7	RC6	—	RC4	RC3	RC2	RC1	RC0	235
0Fh	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	241
010h	PORTE	RE7	RE6	RE5	RE4	RE3	—	RE1	RE0	248
011h	PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	248
012h	TRISA	TRISA7	TRISA6	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	222
013h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	229
014h	TRISC	TRISC7	TRISC6	—	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	235
015h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	241
016h	TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	—	TRISE1	TRISE0	248
017h	TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	255
018h	LATA	LATA7	LATA6	—	LATA4	LATA3	LATA2	LATA1	LATA0	223
019h	LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	230
01Ah	LATC	LATC7	LATC6	—	LATC4	LATC3	LATC2	LATC1	LATC0	236
01Bh	LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	242
01Ch	LATE	LATE7	LATE6	LATE5	LATE4	LATE3	—	LATE1	LATE0	249
01Dh	LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	256
01Eh	—	Unimplemented								
01Fh	ADCPCON0	ADCPON	—	—	—	—	—	—	ADCPRDY	324
08Ch	ADLTHL	LTH								321
08Dh	ADLTHH	LTH								321
08Eh	ADUTHL	UTH								322
08Fh	ADUTHH	UTH								322
090h	ADERRL	ERR								321
091h	ADERRH	ERR								321
092h	ADSTPTL	STPT								320
093h	ADSTPTH	STPT								320
094h	ADFLTRL	FLTR								315
095h	ADFLTRH	FLTR								315
096h	ADACCL	ACC								319
097h	ADACCH	ACC								319
098h	ADACCU	—	—	—	—	—	—	ACC<17:16>		319
099h	ADCNT	CNT								315
09Ah	ADRPT	RPT								314
09Bh	ADPREVL	PREV								318
09Ch	ADPREVH	PREV								318
09Dh	ADRESL	RES								316
09Eh	ADRESH	RES								316
09Fh	ADPCH	—	—	ADPCH<5:0>						311
10Ch	ADACQL	ACQ<7:0>								313
10Dh	ADACQH	—	—	—	ACQ<4:0>					313
10Eh	ADCAP	—	—	—	ADCAP<4:0>					314
10Fh	ADPREL	PRE<7:0>								312

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

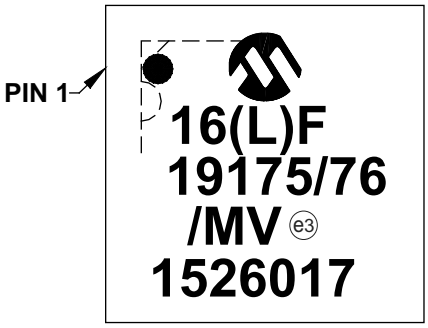
Note 1: Unimplemented data memory locations, read as '0'.

42.1 Package Marking Information (Continued)

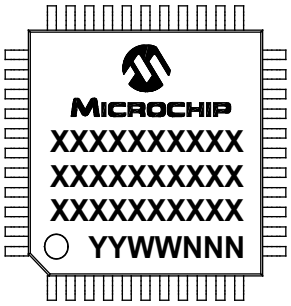
40-Lead UQFN (5x5x0.5 mm)



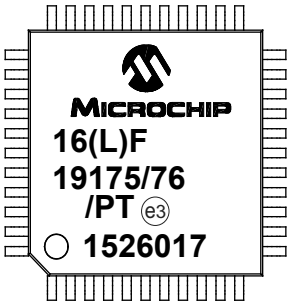
Example



44-Lead TQFP (10x10x1 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
		Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.