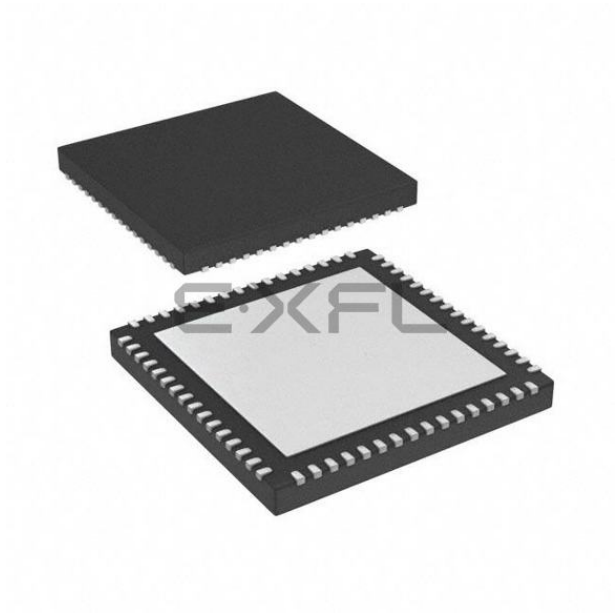


Welcome to E-XFL.COM

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	59
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 45x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19196-i-mr

TABLE 3: 64-PIN ALLOCATION TABLE (PIC16(L)F19195/6/7) (CONTINUED)

I/O(2)	64-Pin TQFP/QFN	ADC	Reference	Comparator	Zero-Cross Detect	DAC	Timers/SMT	CCP	PWM	CWG	MSSP	EUSART	CLC	RTCC	LCD	Interrupt-on-Change	High Current	Pull-up	Basic
RG7	19	ANG7	—	—	—	—	—	—	—	—	—	—	—	—	SEG15 COM7	—	—	Y	—
RH0	26	—	—	—	—	—	—	—	—	—	—	—	—	—	COM4	—	—	Y	—
RH1	25	ADCACT(1)	—	—	—	—	—	—	—	—	—	—	—	—	COM5	—	—	Y	—
RH2	57	—	—	—	—	—	—	—	—	—	—	—	—	—	SEG32 CFLY1	—	—	Y	—
RH3	56	—	—	—	—	—	—	—	—	—	—	—	—	—	SEG40 CFLY2	—	—	Y	—
VLCD3	64	—	—	—	—	—	—	—	—	—	—	—	—	—	VLCD3	—	—	—	—
VDD	10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VDD	38	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	V_{SS}
VSS	41	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VSS
OUT(2)	—	ADGRDA ADGRDB	—	C1OUT C2OUT	—	—	TMR0	CCP1 CCP2	PWM3 PWM4	CWG1A CWG1B CWG1C CWG1D	SDO SCK SCL SDA	TX1 DT1 CK1 TX2 DT2 CK2	CLC1OUT	RTCC	—	—	—	—	—

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.

2: All digital output signals shown in this row are PPS remappable. These signals may be mapped to output onto one or more PORTx pin options.

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. PPS assignments to the other pins 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.

5: These are alternative I^2C logic levels pins.

6: In I^2C logic levels configuration, these pins can operate as either SCL or SDA pins.

2.3 Master Clear ($\overline{MCLR}$) Pin

The $\overline{MCLR}$ pin provides three specific device functions:

- Device Reset (when MCLRE = 1)
- Digital input pin (when MCLRE = 0)
- Device Programming and Debugging

If programming and debugging are not required in the end application then either set the MCLRE Configuration bit to '1' and use the pin as a digital input or clear the MCLRE Configuration bit and leave the pin open to use the internal weak pull-up. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{MCLR}$ pin. Consequently, specific voltage levels (V_{IH} and V_{IL}) and fast signal transitions must not be adversely affected. Therefore, the programmer $\overline{MCLR}/V_{PP}$ output should be connected directly to the pin so that R1 isolates the capacitor, C1 from the $\overline{MCLR}$ pin during programming and debugging operations.

Any components associated with the $\overline{MCLR}$ pin should be placed within 0.25 inch (6 mm) of the pin.

2.4 ICSP™ Pins

The ICSPCLK and ICSPDAT pins are used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of ohms, not to exceed 100Ω.

Pull-up resistors, series diodes and capacitors on the ICSPCLK and ICSPDAT pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be isolated from the programmer by resistors between the application and the device pins or removed from the circuit during programming. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and pin input voltage high (VIH) and input low (VIL) requirements.

For device emulation, ensure that the "Communication Channel Select" (i.e., ICSPCLK/ICSPDAT pins), programmed into the device, matches the physical connections for the ICSP to the Microchip debugger/ emulator tool.

For more information on available Microchip development tools connection requirements, refer to **Section 41.0 "Development Support"**.

4.3.3 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes of the data banks 0-59 and 100 bytes of the data banks 60-63, after the core registers.

The SFRs associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

4.3.4 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank.

TABLE 4-4: GENERAL PURPOSE RAM SIZE AND BANK LOCATION

	PIC16(L)F19195		PIC16(L)F19196		PIC16(L)F19197	
Bank	Address	Size (Bytes)	Address	Size (Bytes)	Address	Size (Bytes)
0	020h-07Fh	96	020h-07Fh	96	020h-07Fh	96
1	0A0h-0EFh	80	0A0h-0EFh	80	0A0h-0EFh	80
2	120h-16Fh	80	120h-16Fh	80	120h-16Fh	80
3	1A0h-1EFh	80	1A0h-1EFh	80	1A0h-1EFh	80
4	220h-26Fh	80	220h-26Fh	80	220h-26Fh	80
5	2A0h-2EFh	80	2A0h-2EFh	80	2A0h-2EFh	80
6	320h-36Fh	80	320h-36Fh	80	320h-36Fh	80
7	3A0h-3EFh	80	3A0h-EFh	80	3A0h-3EFh	80
8	420h-46Fh	80	420h-46Fh	80	420h-46Fh	80
9	4A0h-4EFh	80	4A0h-4EFh	80	4A0h-4EFh	80
10	520h-560h	80	520h-560h	80	520h-560h	80
11	5A0h-5EFh	80	5A0h-5EFh	80	5A0h-5EFh	80
12	620h-64Fh	48	620h-64Fh	80	620h-64Fh	80
13			6A0h-6EFh	80	6A0h-6EFh	80
14			720h-76Fh	80	720h-76Fh	80
15			7A0h-7EFh	80	7A0h-7EFh	80
16			820h-96Fh	80	820h-96Fh	80
17			8A0h-8EFh	80	8A0h-8EFh	80
18			920h-96Fh	80	920h-96Fh	80
19			9A0h-9EFh	80	9A0h-9EFh	80
20			A20h-A6Fh	80	A20h-A6Fh	80
21			AA0h-AEFh	80	AA0h-AEFh	80
22			B20h-B6Fh	80	B20h-B6Fh	80
23			BA0h-BEFh	80	BA0h-BEFh	80
24			C20h-C6Fh	80	C20h-C6Fh	80
25			CA0h-CBFh	32	CA0h-CEFh	80
26					D20h-D6Fh	80
27					DA0h-DEFh	80

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19195/6/7 (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 59											
CPU CORE REGISTERS; see Table 4-3 for specifics											
1D8Ch — 1D9Fh	—	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'.

REGISTER 10-16: PIR5: PERIPHERAL INTERRUPT REQUEST REGISTER 5

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	R/W/HS-0/0
CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF
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	HS = Hardware set

bit 7 **CLC4IF:** CLC4 Interrupt Flag bit
1 = A CLC4OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC4 interrupt event has occurred

bit 6 **CLC3IF:** CLC3 Interrupt Flag bit
1 = A CLC3OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC3 interrupt event has occurred

bit 5 **CLC2IF:** CLC2 Interrupt Flag bit
1 = A CLC2OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC2 interrupt event has occurred

bit 4 **CLC1IF:** CLC1 Interrupt Flag bit
1 = A CLC1OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC1 interrupt event has occurred

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = The Timer1 Gate has gone inactive (the acquisition is complete)
0 = The Timer1 Gate has not gone inactive

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

13.4.2 NVM UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the NVM from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- PFM Row Erase
- Load of PFM write latches
- Write of PFM write latches to PFM memory
- Write of PFM write latches to User IDs

The unlock sequence consists of the following steps and must be completed in order:

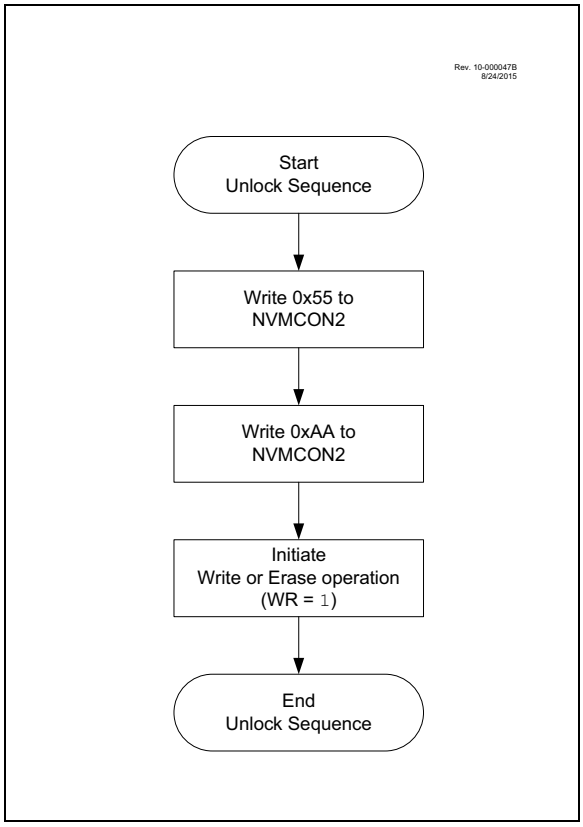
- Write 55h to NVMCON2
- Write AAh to NVMCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

Note:	The two NOP instructions after setting the WR bit that were required in previous devices are not required for PIC16(L)F19195/6/7 devices. See Figure 13-2.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 13-2: NVM UNLOCK SEQUENCE FLOWCHART



EXAMPLE 13-2: NVM UNLOCK SEQUENCE

```
BCF      INTCON, GIE      ; Recommended so sequence is not interrupted
BANKSEL  NVMCON1          ;
BSF      NVMCON1, WREN    ; Enable write/erase
MOVLW    55h              ; Load 55h
MOVWF    NVMCON2          ; Step 1: Load 55h into NVMCON2
MOVLW    AAh              ; Step 2: Load W with AAh
MOVWF    NVMCON2          ; Step 3: Load AAH into NVMCON2
BSF      NVMCON1, WR      ; Step 4: Set WR bit to begin write/erase
BSF      INTCON, GIE      ; Re-enable interrupts
```

Note 1: Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown.

2: Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

TABLE 14-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	212
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	212
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	213
ANSELB	ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	213
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	214
ODCONB	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	214
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0	215
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2	INLVLB1	INLVLB0	215

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTB.

REGISTER 14-38: SLRCONE: PORTE SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
SLRE7	SLRE6	SLRE5	SLRE4	SLRE3	—	SLRE1	SLRE0
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 7-3 **SLRE<7:3>:** PORTE Slew Rate Enable bits
For RE<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **SLRE<1:0>:** PORTE Slew Rate Enable bits
For RE<1:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 14-39: INLVLE: PORTE INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
INLVLE7	INLVLE6	INLVLE5	INLVLE4	INLVLE3	—	INLVLE1	INLVLE0
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 7-3 **INLVLE<7:3>:** PORTE Input Level Select bits
For RE<7:3> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

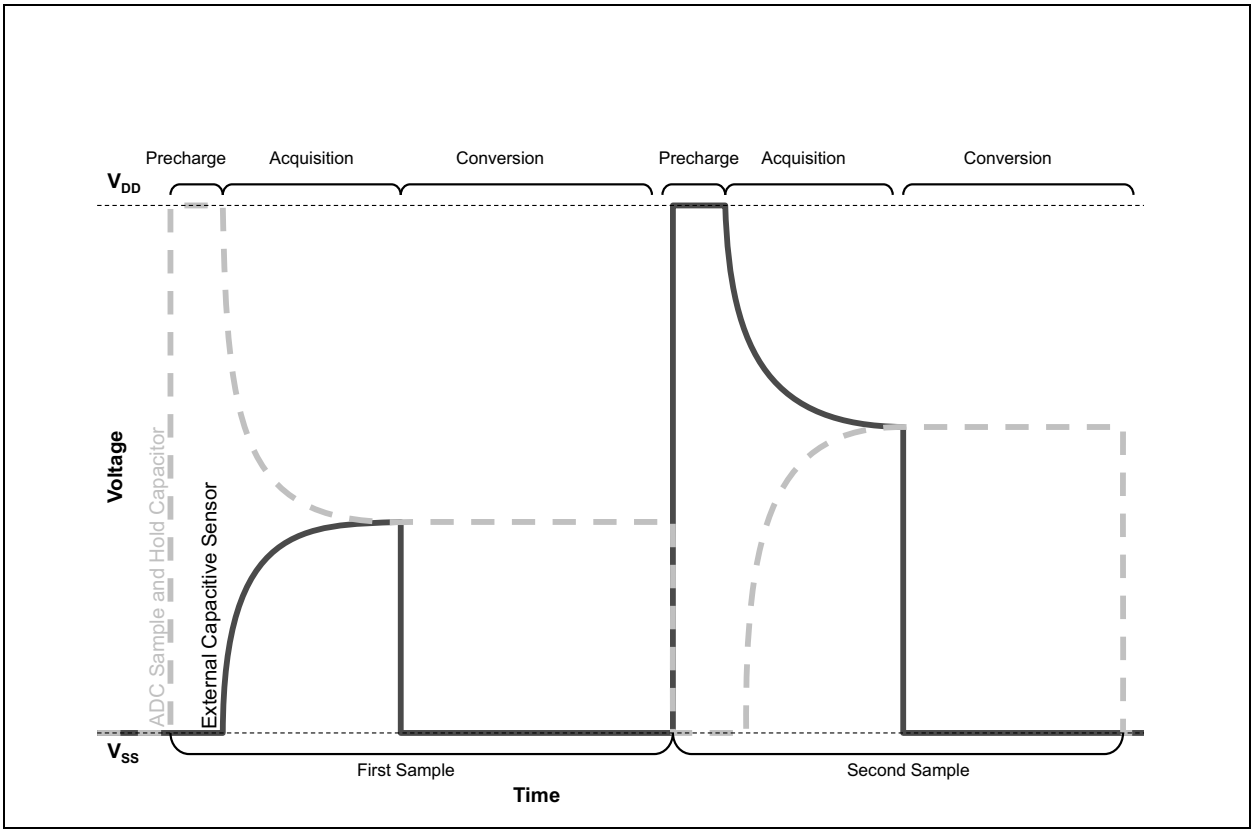
bit 2 **Unimplemented:** Read as '0'

bit 1-0 **INLVLE<1:0>:** PORTE Input Level Select bits
For RE<1:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

19.4.1 CVD OPERATION

A CVD operation begins with the ADC's internal sample and hold capacitor (C_{HOLD}) being disconnected from the path which connects it to the external capacitive sensor node. While disconnected, C_{HOLD} is precharged to V_{DD} or V_{SS}, while the path to the sensor node is precharged to the level opposite that of C_{HOLD}. When the precharge phase is complete, the V_{DD}/V_{SS} precharge paths for the two nodes are shut off and C_{HOLD} and the path to the external sensor node are re-connected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged C_{HOLD} and sensor nodes, which results in a final voltage level setting on C_{HOLD}, which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on C_{HOLD}. This process is then repeated with inverted precharge levels for both the C_{HOLD} and external sensor nodes. Figure 19-7 shows the waveform for two inverted CVD measurements, which is known as differential CVD measurement.

FIGURE 19-7: DIFFERENTIAL CVD MEASUREMENT WAVEFORM



REGISTER 19-27: ADSTPTH: ADC THRESHOLD SETPOINT REGISTER HIGH

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
STPT<15:8>							
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 **STPT<15:8>**: ADC Threshold Setpoint MSB. Upper byte of ADC threshold setpoint, depending on ADCALC, may be used to determine ERR, see Register 19-29 for more details.

REGISTER 19-28: ADSTPTL: ADC THRESHOLD SETPOINT REGISTER LOW

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
STPT<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 **STPT<7:0>**: ADC Threshold Setpoint LSB. Lower byte of ADC threshold setpoint, depending on ADCALC, may be used to determine ERR, see Register 19-30 for more details.

REGISTER 19-29: ADERRH: ADC SETPOINT ERROR REGISTER HIGH

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ERR<15:8>							
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 **ERR<15:8>**: ADC Setpoint Error MSB. Upper byte of ADC Setpoint Error. Setpoint Error calculation is determined by ADCALC bits of ADCON3, see Register 19-4 for more details.

TABLE 25-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TMR0L	Holding Register for the Least Significant Byte of the 16-bit TMR0 Register								359*
TMR0H	Holding Register for the Most Significant Byte of the 16-bit TMR0 Register								359*
T0CON0	T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				362
T0CON1	T0CS<2:0>			T0ASYNC	T0CKPS<3:0>				363
T0CKIPPS	—	—	—	—	T0CKIPPS<3:0>				258
TMR0PPS	—	—	—	TMR0PPS<4:0>					258
T1GCON	GE	GPOL	GTM	GSPM	GGO/$\overline{\text{DONE}}$	GVAL	—	—	374
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	147
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	157
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	148

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the Timer0 module.

* Page with Register information.

27.5 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the TxCON register are cleared).
- The diagrams illustrate any clock except Fosc/4 and show clock-sync delays of at least two full cycles for both ON and Timer2/4_ers. When using Fosc/4, the clock-sync delay is at least one instruction period for Timer2/4_ers; ON applies in the next instruction period.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in **Section 29.0 "Capture/Compare/PWM Modules"**. The signals are not a part of the Timer2/4 module.

27.5.1 SOFTWARE GATE MODE

This mode corresponds to legacy Timer2/4 operation. The timer increments with each clock input when ON = 1 and does not increment when ON = 0. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 27-4. With PRx = 5, the counter advances until TMRx = 5, and goes to zero with the next clock.

27.6 Timer2/4 Operation During Sleep

When PSYNC = 1, Timer2/4 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and T2PR registers will remain unchanged while processor is in Sleep mode.

When PSYNC = 0, Timer2/4 will operate in Sleep as long as the clock source selected is also still running. Selecting the LFINTOSC, MFINTOSC, or HFINTOSC oscillator as the timer clock source will keep the selected oscillator running during Sleep.

28.7.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMTxTMR will increment. Upon a falling edge of the external signal, the SMTxCPW register will update to the current value of the SMTxTMR. Example waveforms for both repeated and single acquisitions are provided in Figure 28-4 and Figure 28-5.

FIGURE 28-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

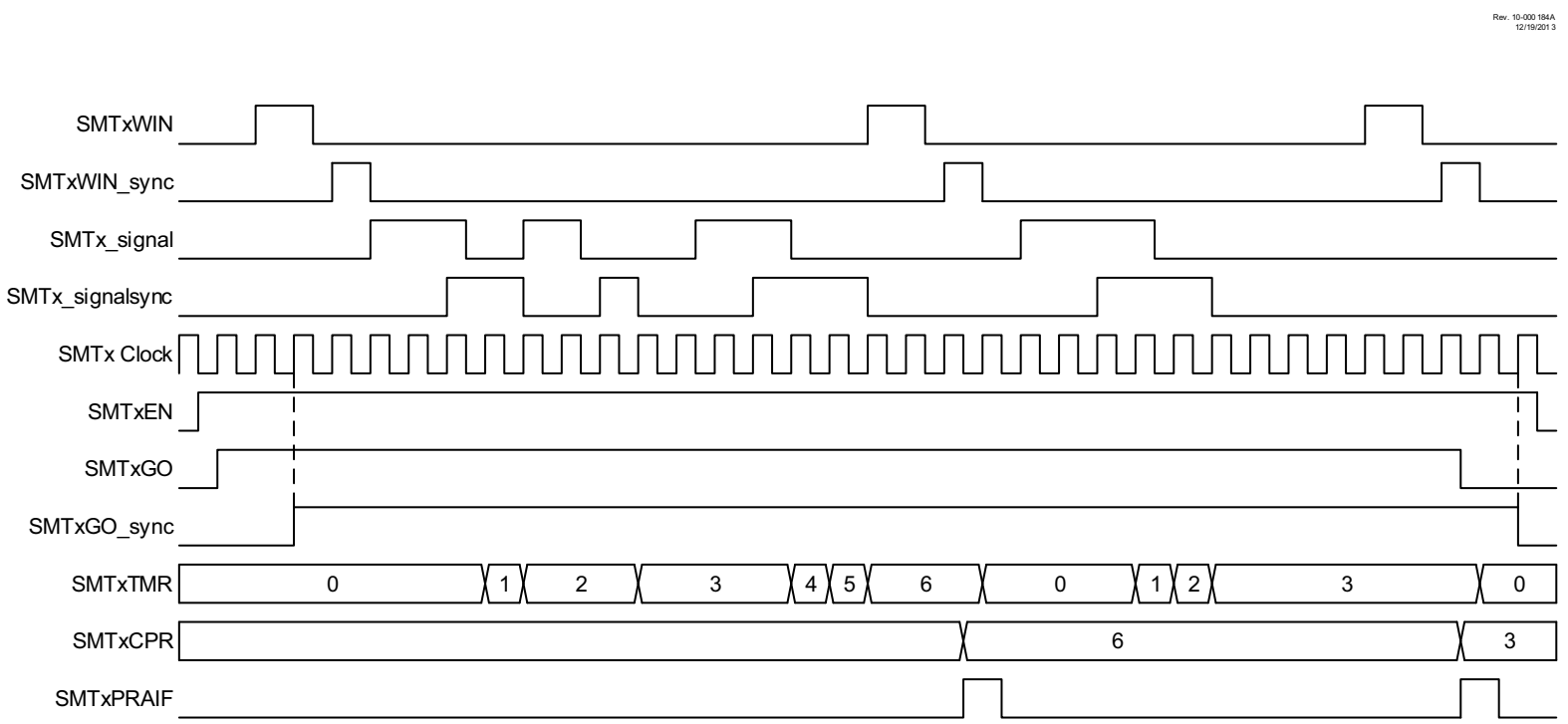
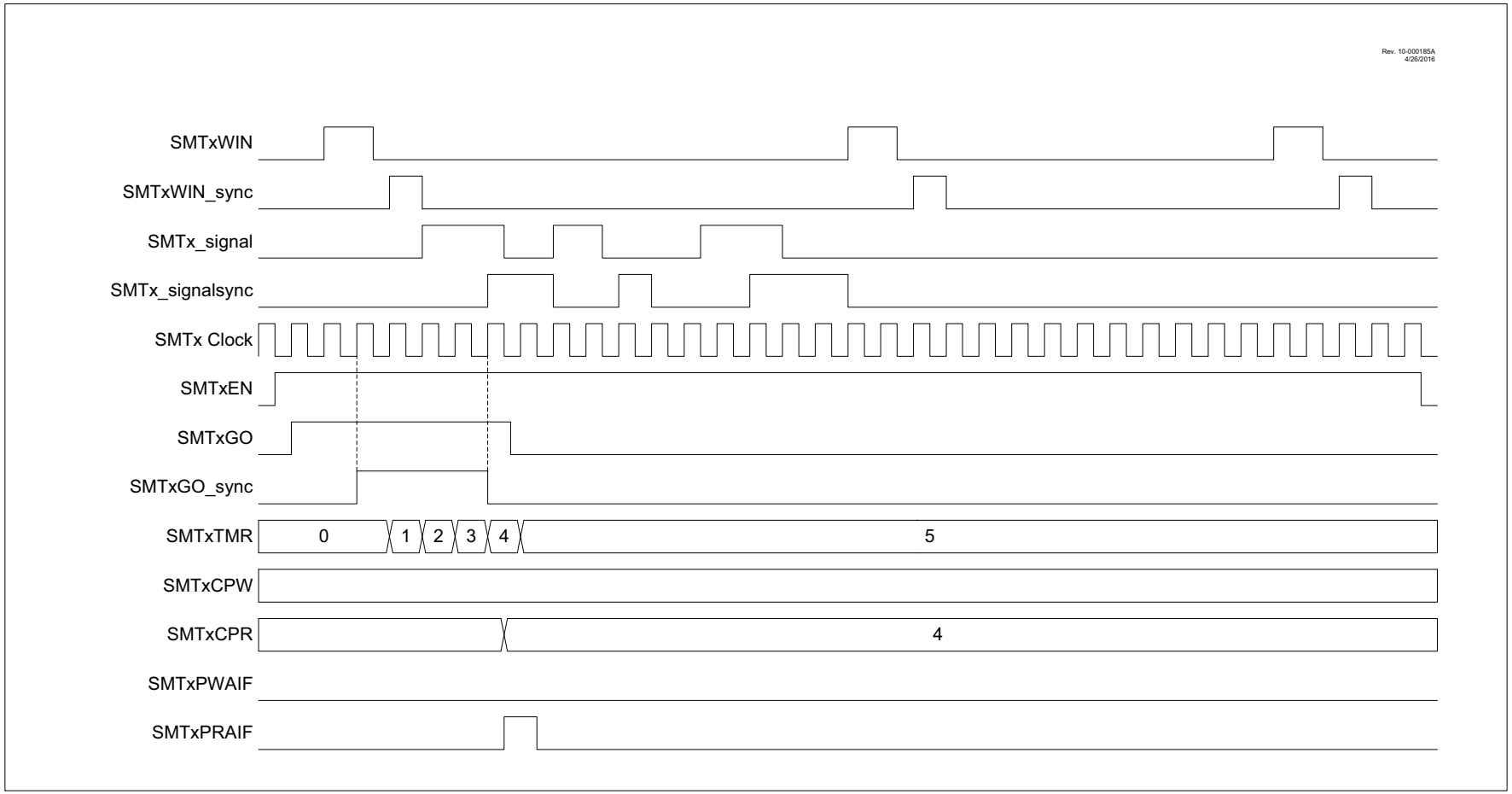


FIGURE 28-15: TIME OF FLIGHT MODE SINGLE ACQUISITION TIMING DIAGRAM



34.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDxCON register selects 16-bit mode.

The SPxBRGH, SPxBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXxSTA register and the BRG16 bit of the BAUDxCON register. In Synchronous mode, the BRGH bit is ignored.

Table 34-1 contains the formulas for determining the baud rate. Example 34-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various Asynchronous modes have been computed for your convenience and are shown in Table 34-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPxBRGH, SPxBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is idle before changing the system clock.

EXAMPLE 34-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$Desired\ Baud\ Rate = \frac{FOSC}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SPxBRGH:SPxBRGL:

$$X = \frac{\frac{FOSC}{Desired\ Baud\ Rate}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

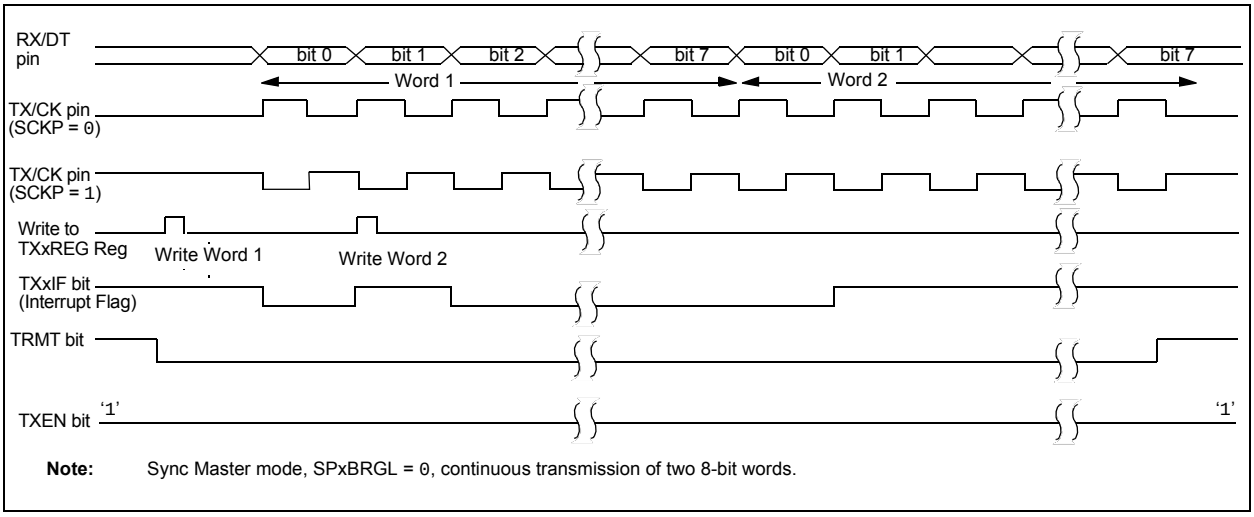
$$Calculated\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$Error = \frac{Calc.\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}$$

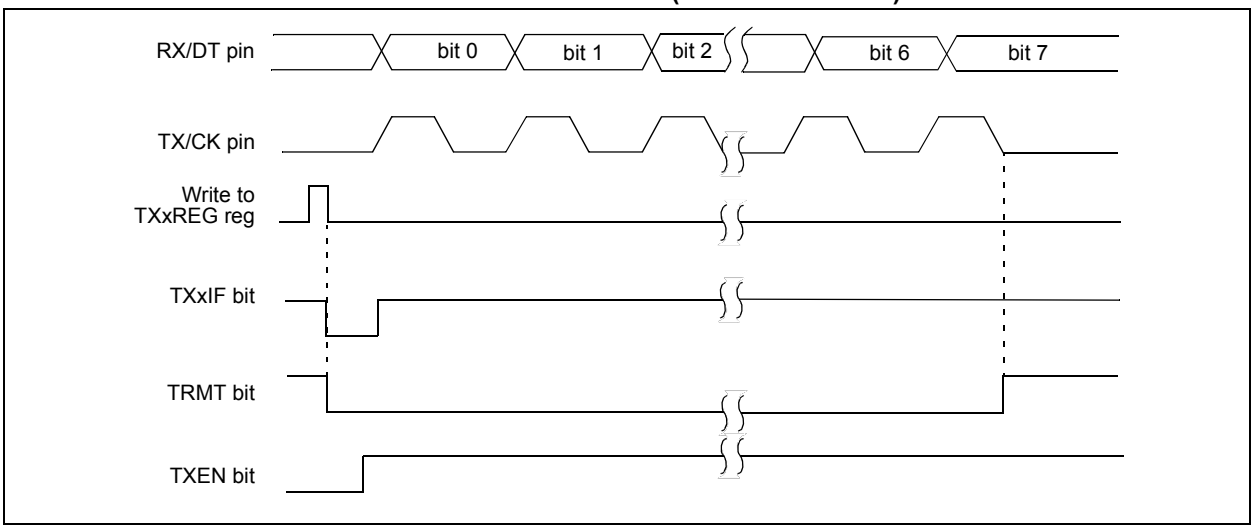
$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

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.

FIGURE 35-14: TYPE-B WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE

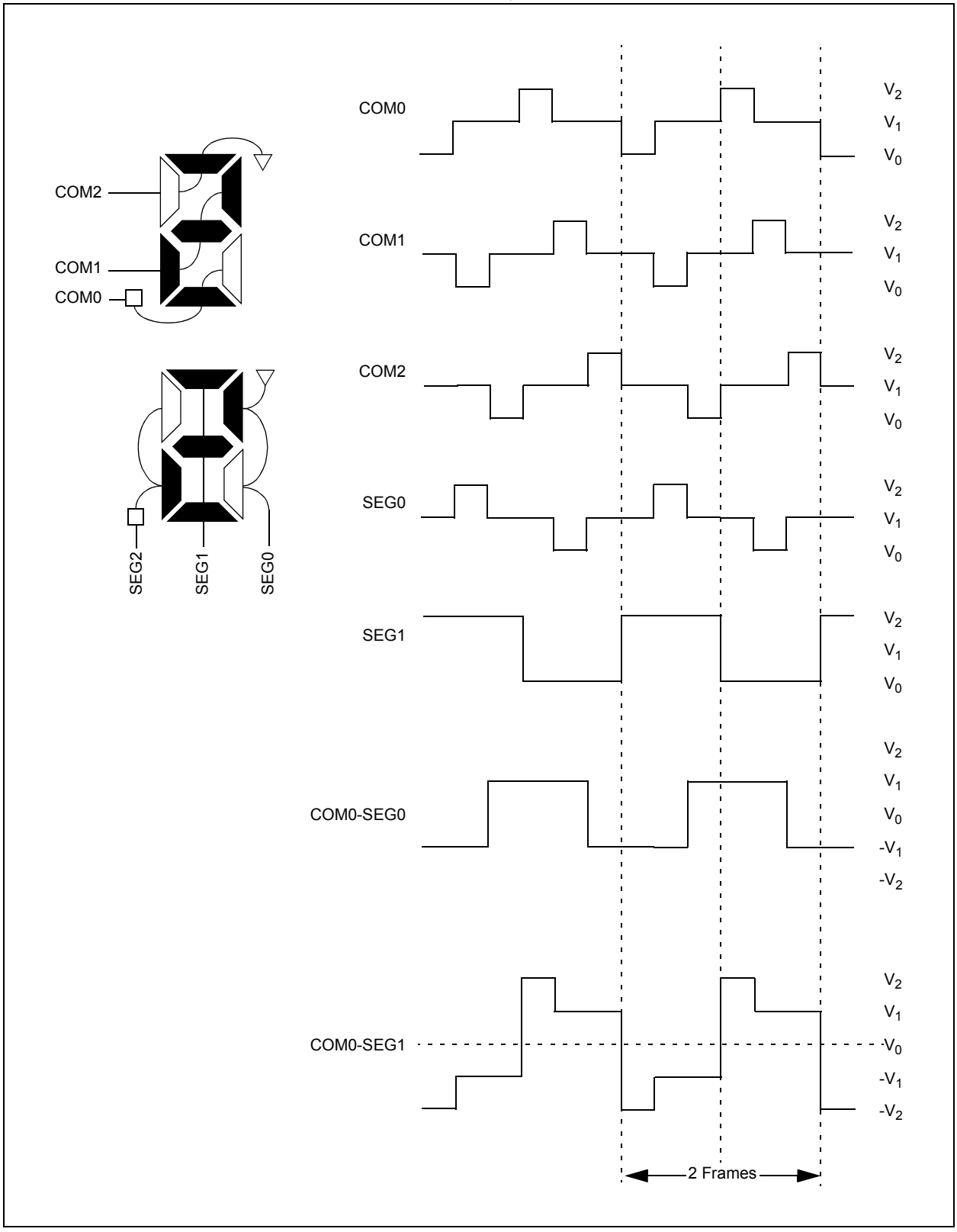


TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19195/6/7 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
68Dh	—	Unimplemented								
68Eh	—	Unimplemented								
68Fh	—	Unimplemented								
690h	—	Unimplemented								
691h	—	Unimplemented								
692h	—	Unimplemented								
693h	—	Unimplemented								
694h	—	Unimplemented								
695h	—	Unimplemented								
696h	—	Unimplemented								
697h	—	Unimplemented								
698h	—	Unimplemented								
699h	—	Unimplemented								
69Ah	—	Unimplemented								
69Bh	—	Unimplemented								
69Ch	—	Unimplemented								
69Dh	—	Unimplemented								
69Eh	—	Unimplemented								
69Fh	—	Unimplemented								
70Ch	PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	157
70Dh	PIR1	OSFIF	CSWIF	—	—	—	—	ADTIF	ADIF	158
70Eh	PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	159
70Fh	PIR3	RC2IF	TX2IF	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	160
710h	PIR4	—	—	—	—	TMR4IF	—	TMR2IF	TMR1IF	161
711h	PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	162
712h	PIR6	CRIF	—	—	—	—	—	CCP2IF	CCP1IF	163
713h	PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	164
714h	PIR8	LCDIF	RTCCIF	—	—	—	SMT1PWAIF	SMT1PRAIF	SMT1IF	165
715h	—	Unimplemented								
716h	PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	148
717h	PIE1	OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE	149
718h	PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	150
719h	PIE3	RC2IE	TX2IE	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	151
71Ah	PIE4	—	—	—	—	TMR4IF	—	TMR2IE	TMR1IE	152
71Bh	PIE5	CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	153
71Ch	PIE6	CRIE	—	—	—	—	—	CCP2IE	CCP1IE	154
71Dh	PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	155
71Eh	PIE8	LCDIE	RTCCIE	—	—	—	SMT1PWAIE	SMT1PRAIE	SMT1IE	156
71Fh	—	Unimplemented								

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