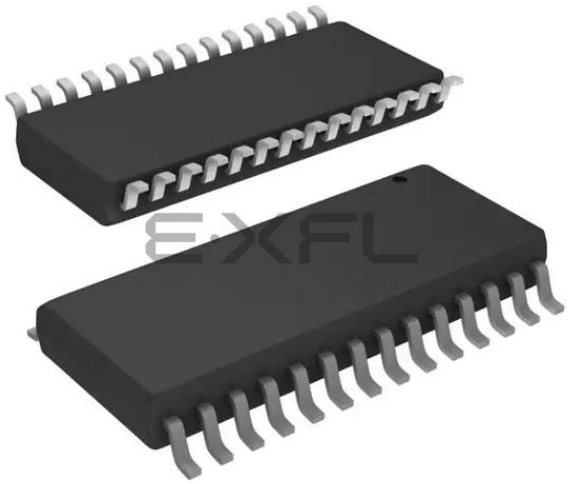


Welcome to E-XFL.COM

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)	1.8V ~ 3.6V
Data Converters	A/D 20x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf19156t-i-so

TABLE 5: 48-PIN ALLOCATION TABLE (PIC16(L)F19185/86)

I/O(2)	48-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
RA0	21	ANA0	—	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	CLCIN0(1)	—	SEG0	IOCA0	—	Y	—
RA1	22	ANA1	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	CLCIN1(1)	—	SEG1	IOCA1	—	Y	—
RA2	23	ANA2	—	C1IN0+ C2IN0+	—	DAC1OUT1	—	—	—	—	—	—	—	—	SEG2	IOCA2	—	Y	—
RA3	24	ANA3	VREF+	C1IN1+	—	DAC1REF+	—	—	—	—	—	—	—	—	SEG3	IOCA3	—	Y	—
RA4	25	ANA4	—	—	—	—	T0CKI(1)	—	—	—	—	—	—	—	SEG4 COM3	IOCA4	—	Y	—
RA5	26	—	—	—	—	—	—	—	—	—	$\overline{SS}$(1)	—	—	—	—	IOCA5	—	Y	VBAT
RA6	33	ANA6	—	—	—	—	—	—	—	—	—	—	—	—	SEG6	IOCA6	—	Y	CLKOUT OSC2
RA7	32	ANA7	—	—	—	—	—	—	—	—	—	—	—	—	SEG7	IOCA7	—	Y	OSC1 CLKIN
RB0	8	ANB0	—	C2IN1+	ZCD	—	—	—	—	CWG1IN(1)	—	—	—	—	SEG8	IOCB0	—	Y	INTPPS
RB1	9	ANB1	—	C1IN3- C2IN3-	—	—	—	—	—	—	SCL, SDA(1, 3, 4, 5, 6)	—	—	—	SEG9	IOCB1	HIB1	Y	—
RB2	10	ANB2	—	—	—	—	—	—	—	—	SCL, SDA(1, 3, 4, 5, 6)	—	—	—	SEG10 CFLY1	IOCB2	—	Y	—
RB3	11	ANB3	—	C1IN2- C2IN2-	—	—	—	—	—	—	—	—	—	—	SEG11 CFLY2	IOCB3	—	Y	—
RB4	16	ANB4 ADCACT(1)	—	—	—	—	—	—	—	—	—	—	—	—	COM0	IOCB4	—	Y	—
RB5	17	ANB5	—	—	—	—	T1G(1)	—	—	—	—	—	—	—	SEG13 COM1	IOCB5	—	Y	—
RB6	18	ANB6	—	—	—	—	—	—	—	—	—	TX2(1) CK2(1)	CLCIN2(1)	—	SEG14	IOCB6	—	Y	ICDCLK/ ICSPCLK
RB7	19	ANB7	—	—	—	DAC1OUT2	—	—	—	—	—	RX2(1) DT2(1)	CLCIN3(1)	—	SEG15	IOCB7	—	Y	ICDDAT/ ICSPDAT

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 INLCVL 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 and SDA pins.

© 2017 Microchip Technology Inc.

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 19											
CPU CORE REGISTERS; see Table 4-3 for specifics											
98Ch	—	Unimplemented								---- ----	---- ----
98Dh	—	Unimplemented								---- ----	---- ----
98Eh	—	Unimplemented								---- ----	---- ----
98Fh	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	0000 0000	0000 0000
990h	CM1CON0	ON	OUT	—	POL	—	—	HYS	SYNC	00-0 --00	00-0 --00
991h	CM1CON1	—	—	—	—	—	—	INTP	INTN	0000 0000	0000 0000
992h	CM1NSEL	—	—	—	—	—	NCH<2:0>			0000 0000	0000 0000
992h		—	—	—	—	—	NCH2	NCH1	NCH0	0000 0000	0000 0000
993h	CM1PSEL	—	—	—	—	PCH<3:0>				0000 0000	0000 0000
993h		—	—	—	—	—	PCH2	PCH1	PCH0	0000 0000	0000 0000
994h	CM2CON0	ON	OUT	—	POL	—	—	HYS	SYNC	00-0 --00	00-0 --00
995h	CM2CON1	—	—	—	—	—	—	INTP	INTN	0000 0000	0000 0000
996h	CM2NSEL	—	—	—	—	—	NCH<2:0>			0000 0000	0000 0000
996h		—	—	—	—	—	NCH2	NCH1	NCH0	0000 0000	0000 0000
997h	CM2PSEL	—	—	—	—	—	PCH<2:0>			0000 0000	0000 0000
997h		—	—	—	—	—	PCH2	PCH1	PCH0	0000 0000	0000 0000
998h	—	Unimplemented								---- ----	---- ----
999h	—	Unimplemented								---- ----	---- ----
99Ah	—	Unimplemented								---- ----	---- ----
99Bh	—	Unimplemented								---- ----	---- ----
99Ch	—	Unimplemented								---- ----	---- ----
99Dh	—	Unimplemented								---- ----	---- ----
99Eh	—	Unimplemented								---- ----	---- ----
99Fh	—	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'.

© 2017 Microchip Technology Inc.

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'.

© 2017 Microchip Technology Inc.

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)											
1F4Eh	ANSELC	ANSC7	ANSC6	—	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	1111 1111	1111 1111
1F4Fh	WPUC	WPUC7	WPUC6	—	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	0000 0000	0000 0000
1F50h	ODCONC	ODCC7	ODCC6	—	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	0000 0000	0000 0000
1F51h	SLRCONC	SLRC7	SLRC6	—	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111
1F52h	INLVLC	INLVLC7	INLVLC6	—	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
1F53h	IOCCP	IOCCP7	IOCCP6	—	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
1F54h	IOCCN	IOCCN7	IOCCN6	—	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
1F55h	IOCCF	IOCCF7	IOCCF6	—	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000
1F56h	—	Unimplemented								---- ----	---- ----
1F57h	—	Unimplemented								---- ----	---- ----
1F58h	—	Unimplemented								---- ----	---- ----
1F59h	ANSELD	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	1111 1111	1111 1111
1F5Ah	WPUD	WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0	0000 0000	0000 0000
1F5Bh	ODCOND	ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0	0000 0000	0000 0000
1F5Ch	SLRCOND	SLRD7	SLRD6	SLRD5	SLRD4	SLRD3	SLRD2	SLRD1	SLRD0	1111 1111	1111 1111
1F5Dh	INLVLD	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1	INLVLD0	1111 1111	1111 1111
1F5Eh	—	Unimplemented								---- ----	---- ----
1F5Fh	—	Unimplemented								---- ----	---- ----
1F60h	—	Unimplemented								---- ----	---- ----
1F61h	—	Unimplemented								---- ----	---- ----
1F62h	—	Unimplemented								---- ----	---- ----
1F63h	—	Unimplemented								---- ----	---- ----

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

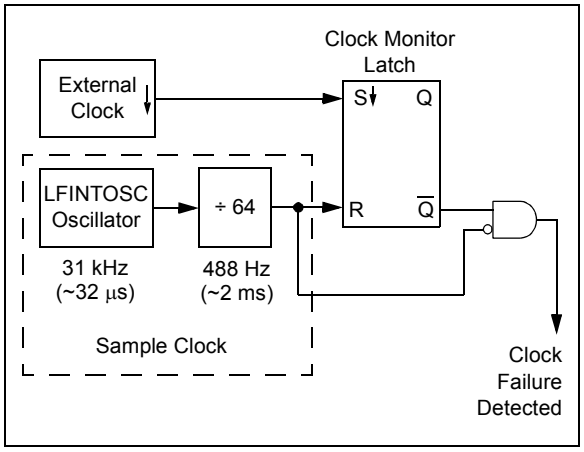
Note 1: Unimplemented data memory locations, read as '0'.

© 2017 Microchip Technology Inc.

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.

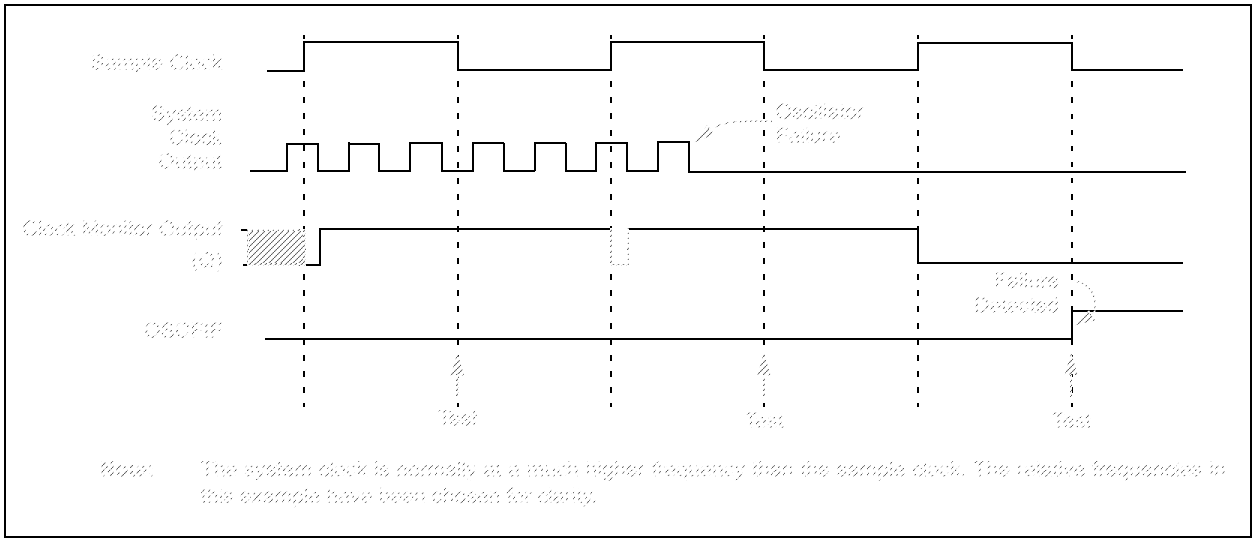
9.4.3 FAIL-SAFE CONDITION CLEARING

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.

REGISTER 14-37: WPUE: WEAK PULL-UP PORTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
WPUE7	WPUE6	WPUE5	WPUE4	WPUE3	—	WPUE1	WPUE0
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 **WPUE<7:3>**: Weak Pull-up Register bits(1)
1 = Pull-up enabled
0 = Pull-up disabled

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

bit 1-0 **WPUE<1:0>**: Weak Pull-up Register bits(1)
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: The weak pull-up device is automatically disabled if the pin is configured as an output.

REGISTER 14-38: ODCONE: PORTE OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ODCE7	ODCE6	ODCE5	ODCE4	ODCE3	—	ODCE1	ODCE0
bit 7							bit 0

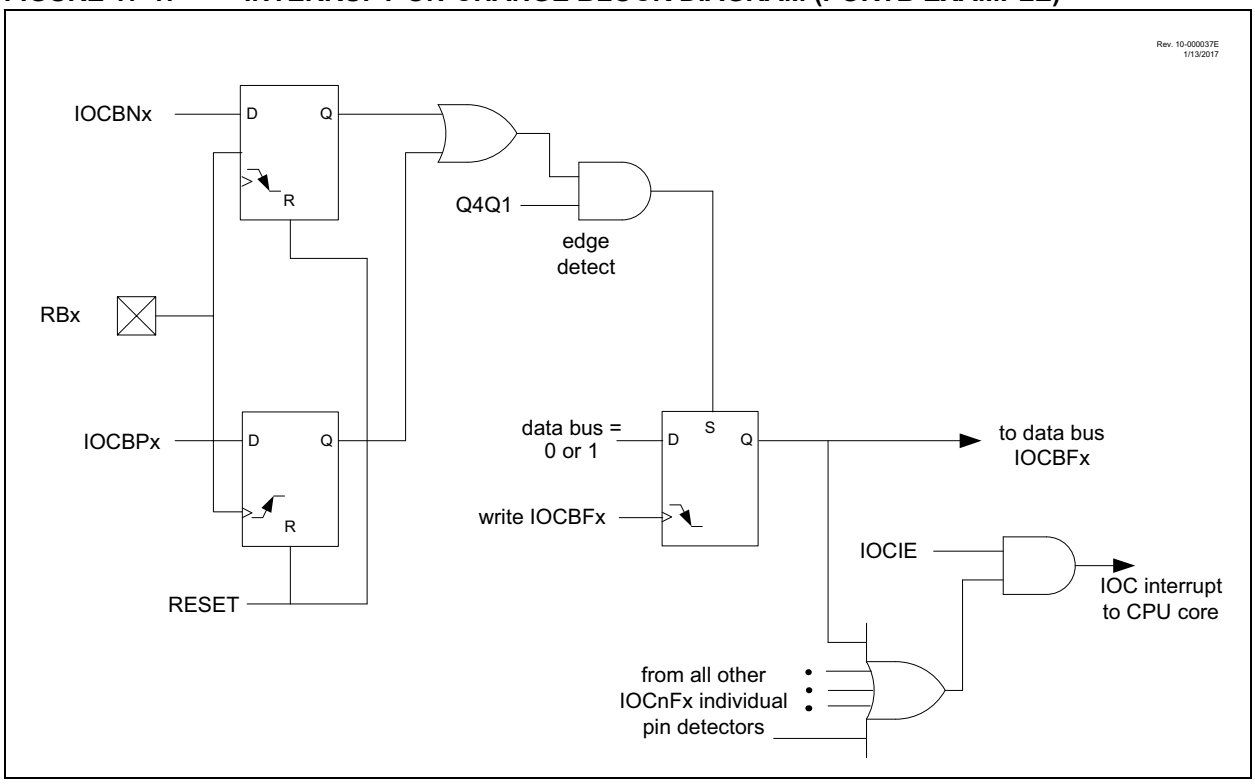
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-3 **ODCE<7:3>:** PORTE Open-Drain Enable bits
For RE<7:3> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

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

bit 1-0 **ODCE<1:0>:** PORTE Open-Drain Enable bits
For RE<1:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

FIGURE 17-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTB EXAMPLE)



REGISTER 19-11: ADACQL: ADC ACQUISITION TIME CONTROL REGISTER (LOW BYTE)

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
ACQ<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 ACQ<7:0>: Acquisition (charge share time) Select bits
See Table 19-6.

REGISTER 19-12: ADACQH: ADC ACQUISITION TIME CONTROL REGISTER (HIGH BYTE)

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ACQ<12:8>				
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 7-5 **Unimplemented**: Read as ‘0’

bit 4-0 ACQ<12:8>: Acquisition (charge share time) Select bits
See Table 19-6.

TABLE 19-5: ACQUISITION TIME

ADACQ	Acquisition time
1 1111 1111 1111	8191 clocks of the selected ADC clock
1 1111 1111 1110	8190 clocks of the selected ADC clock
1 1111 1111 1101	8189 clocks of the selected ADC clock
...	...
0 0000 0000 0010	2 clocks of the selected ADC clock
0 0000 0000 0001	1 clock of the selected ADC clock
0 0000 0000 0000	Not included in the data conversion cycle(1)

Note 1: If ADPRE is not equal to ‘0’, then ADACQ = b′0_0000_0000_0000 means Acquisition time is 8192 clocks of the selected ADC clock.

REGISTER 19-33: ADUTHH: ADC UPPER THRESHOLD HIGH BYTE REGISTER

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
UTH<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 **UTH<15:8>**: ADC Upper Threshold MSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 19-34: ADUTHL: ADC UPPER THRESHOLD LOW BYTE REGISTER

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
UTH<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 **UTH<7:0>**: ADC Upper Threshold LSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

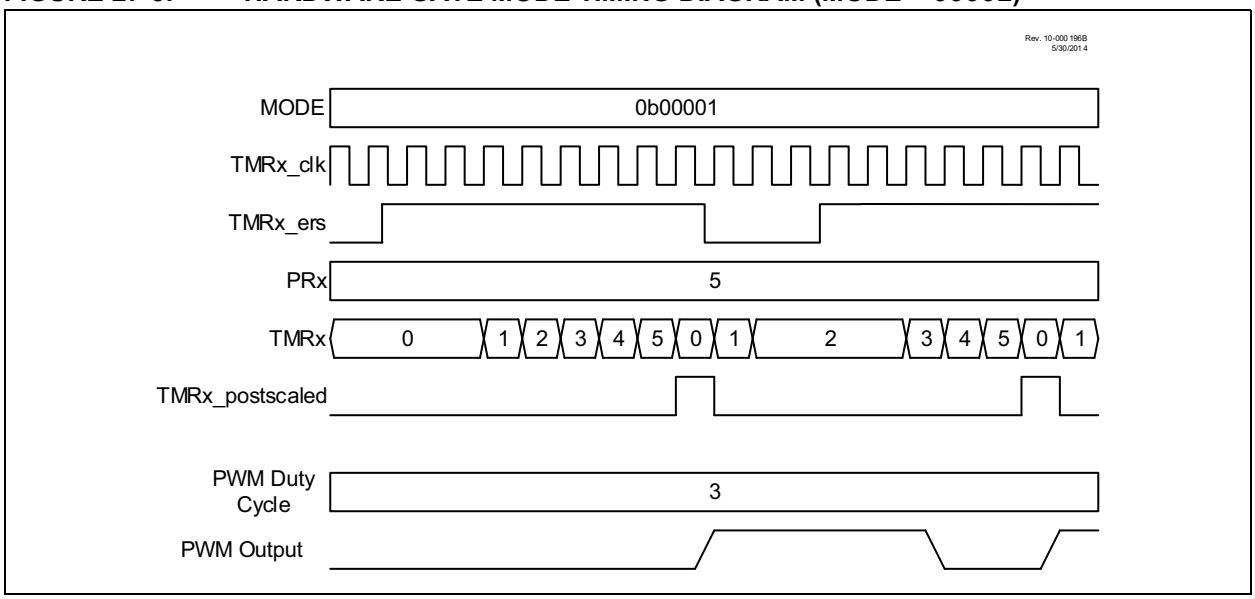
27.5.2 HARDWARE GATE MODE

The Hardware Gate modes operate the same as the Software Gate mode except the TMRx_ers external signal gates the timer. When used with the CCP the gating extends the PWM period. If the timer is stopped when the PWM output is high then the duty cycle is also extended.

When MODE<4:0> = 00001 then the timer is stopped when the external signal is high. When MODE<4:0> = 00010 then the timer is stopped when the external signal is low.

Figure 27-5 illustrates the Hardware Gating mode for MODE<4:0> = 00001 in which a high input level starts the counter.

FIGURE 27-5: HARDWARE GATE MODE TIMING DIAGRAM (MODE = 00001)



27.5.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

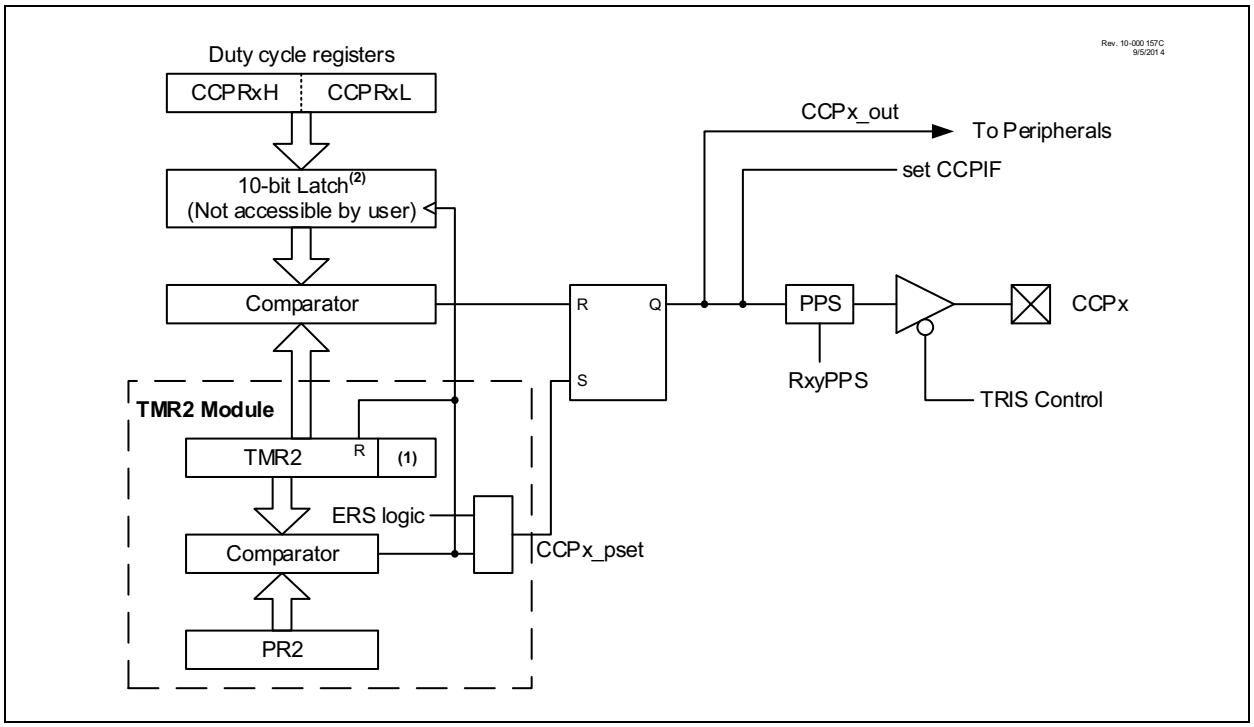
In Level -Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = `01110`)
- High Reset level (MODE<4:0> = `01111`)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.

FIGURE 29-4: SIMPLIFIED PWM BLOCK DIAGRAM



29.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Use the desired output pin RxyPPS control to select CCPx as the source and disable the CCPx pin output driver by setting the associated TRIS bit.
2. Load the PR2 register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRxL register, and the CCPRxH register with the PWM duty cycle value and configure the CCPxFMT bit of the CCPxCON register to set the proper register alignment.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR4 register. See Note below.
   - Configure the CKPS bits of the T2CON register with the Timer prescale value.
   - Enable the Timer by setting the Timer2 ON bit of the T2CON register.
6. Enable PWM output pin:
   - Wait until the Timer overflows and the TMR2IF bit of the PIR4 register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRIS bit.

Note:	In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

29.3.3 CCP/PWM CLOCK SELECTION

The PIC16(L)F19155/56/75/76/85/86 allows each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

33.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

33.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 33-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

33.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100).

When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven.

When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.
2:	When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SS}$ pin control.
3:	While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

33.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 33-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

Note: SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets the CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the ninth SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.

REGISTER 34-7: SPxBRGH(1, 2): BAUD RATE GENERATOR HIGH 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<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 **SPxBRG<15:8>:** Upper eight bits of the Baud Rate Generator

Note 1: SPxBRGH value is ignored for all modes unless BAUDxCON<BRG16> is active.

2: Writing to SPxBRGH resets the BRG counter.

35.5.2 AUTOMATIC POWER MODE SWITCHING

Each segment within an LCD display is perceived electrically like a small capacitor. Due to this fact, power is mainly consumed during the transition periods when voltage is being supplied to the segments. So in order to manage total current consumption, the LCD reference ladder can be used in different power modes during these transition periods. Control of the LCD reference ladder is done through the LCDRL register (see Register 35-7).

The automatic power switching using Type-A/Type-B, can optimize the power consumption for a given contrast.

As shown in Figure 35-3, Power Mode A is active for a programmable time, beginning when the LCD segment waveform is transitioning. The LRLAT<2:0> bits (LCDRL<2:0>) select how long Mode A is active. Power mode B is active for the remaining time before the segments or commons change again.

As shown in Figure 35-3, there are 32 counts in a single segment time. Type-A is used when the wave form is in transition. Type-B can be used when the segment voltage is stable or not in transition.

FIGURE 35-3: LCD REFERENCE LADDER POWER MODE SWITCHING DIAGRAM

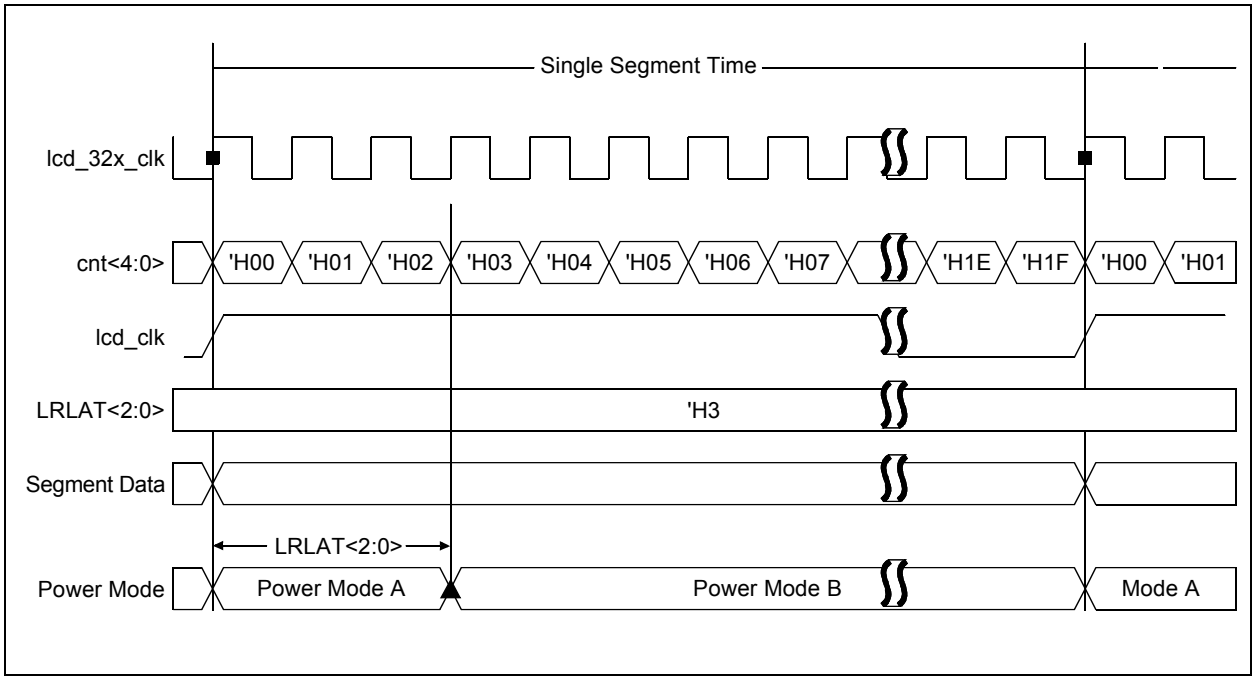


TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
30Ch	CCPR1L	RL								460
30Dh	CCPR1H	RH								461
30Eh	CCP1CON	CCP1EN	—	CCP1OUT	CCP1FMT	CCP1MODE<3:0>				458
		—		—	—	CCP1-MODE3	CCP1-MODE2	CCP1-MODE1	CCP1MODE0	458
30Fh	CCP1CAP	CCP1CTS								460
		—	—	—	—	—	CCP1CTS2	CCP1CTS1	CCP1CTS0	460
310h	CCPR2L	RL								460
311h	CCPR2H	RH								461
312h	CCP2CON	CCP2EN	—	CCP2OUT	CCP2FMT	CCP2MODE<3:0>				458
		—		—	—	CCP2-MODE3	CCP2-MODE2	CCP2-MODE1	CCP2MODE0	458
313h	CCP2CAP	CCP2CTS								460
		—	—	—	—	—	CCP2CTS2	CCP2CTS1	CCP2CTS0	460
314h	PWM3DCL	PWM3DC<1:0>		—	—	—	—	—	—	468
		PWM3DC1	PWM3DC0	—	—	—	—	—	—	468
315h	PWM3DCH	PWM3DC								468
		PWM3DC9	PWM3DC8	PWM3DC7	PWM3DC6	PWM3DC5	PWM3DC4	PWM3DC3	PWM3DC2	468
316h	PWM3CON	PWM3EN	—	PWM3OUT	PWM3POL	—	—	—	—	467
317h	—	Unimplemented								467
318h	PWM4DCL	PWM4DC<1:0>		—	—	—	—	—	—	468
		PWM4DC1	PWM4DC0	—	—	—	—	—	—	468
319h	PWM4DCH	PWM4DC								468
		PWM4DC9	PWM4DC8	PWM4DC7	PWM4DC6	PWM4DC5	PWM4DC4	PWM4DC3	PWM4DC2	468
31Ah	PWM4CON	PWM4EN	—	PWM4OUT	PWM4POL	—	—	—	—	467
31Bh 31Fh	—	Unimplemented								
38Ch	—	Unimplemented								
38Dh	—	Unimplemented								
38Eh	—	Unimplemented								
38Fh	—	Unimplemented								
390h	—	Unimplemented								
391h	—	Unimplemented								
392h	—	Unimplemented								
393h	—	Unimplemented								
394h	—	Unimplemented								
395h	—	Unimplemented								
396h	—	Unimplemented								
397h	—	Unimplemented								
398h	—	Unimplemented								
399h	—	Unimplemented								
39Ah	—	Unimplemented								
39Bh	—	Unimplemented								
39Ch	—	Unimplemented								
39Dh	—	Unimplemented								
39Eh	—	Unimplemented								
39Fh	—	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'.

© 2017 Microchip Technology Inc.

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
1F33h	—	Unimplemented								
1F34h	—	Unimplemented								
1F35h	—	Unimplemented								
1F36h	—	Unimplemented								
1F37h	—	Unimplemented								
1F38h	ANSELA	ANSA7	ANSA6	—	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	223
1F39h	WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	224
1F3Ah	ODCONA	ODCA7	ODCA6	—	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	224
1F3Bh	SLRCONA	SLRA7	SLRA6	—	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0	225
1F3Ch	INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	225
1F3Dh	IOCAP	IOCAP7	IOCAP6	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	
1F3Eh	IOCAN	IOCAN7	IOCAN6	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	
1F3Fh	IOCAF	IOCAF7	IOCAF6	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	
1F40h	—	Unimplemented								
1F41h	—	Unimplemented								
1F42h	—	Unimplemented								
1F43h	ANSELB	ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	230
1F44h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	231
1F45h	ODCONB	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	231
1F46h	SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0	232
1F47h	INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2	INLVLB1	INLVLB0	232
1F48h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	277
1F49h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	277
1F4Ah	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	277
1F4Bh	—	Unimplemented								
1F4Ch	—	Unimplemented								
1F4Dh	—	Unimplemented								
1F4Eh	—	Unimplemented								
1F4Fh	WPUC	WPUC7	WPUC6	—	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	236
1F50h	ODCONC	ODCC7	ODCC6	—	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	237
1F51h	SLRCONC	SLRC7	SLRC6	—	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	237
1F52h	INLVLC	INLVLC7	INLVLC6	—	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	238
1F53h	IOCCP	IOCCP7	IOCCP6	—	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	278
1F54h	IOCCN	IOCCN7	IOCCN6	—	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	278
1F55h	IOCCF	IOCCF7	IOCCF6	—	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	278
1F56h	—	Unimplemented								
1F57h	—	Unimplemented								
1F58h	—	Unimplemented								
1F59h	ANSELD	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	242
1F5Ah	WPUD	WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0	243
1F5Bh	ODCOND	ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0	243
1F5Ch	SLRCOND	SLRD7	SLRD6	SLRD5	SLRD4	SLRD3	SLRD2	SLRD1	SLRD0	244
1F5Dh	INLVLD	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1	INLVLD0	244

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'.

41.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
  - MPLAB® XPRESS IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

41.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (6/2017)

This is the initial release of the document.