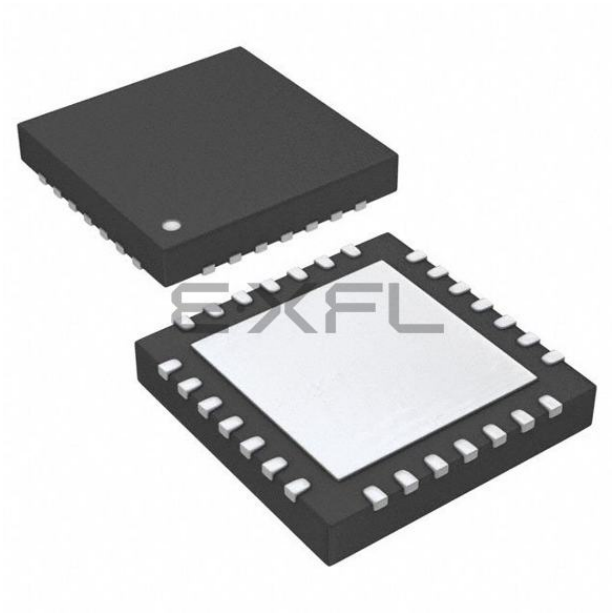


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf24k42-e-ml

TABLE 1-1: DEVICE FEATURES

Features	PIC18(L)F24K42	PIC18(L)F25K42
Program Memory (Bytes)	16384	32768
Program Memory (Instructions)	8192	16384
Data Memory (Bytes)	1024	2048
Data EEPROM Memory (Bytes)	256	256
I/O Ports	A,B,C,E(1)	A,B,C,E(1)
Capture/Compare/PWM Modules (CCP)	4	4
10-Bit Pulse-Width Modulator (PWM)	4	4
12-Bit Analog-to-Digital Module (ADC^2) with Computation Accelerator	5 internal 24 external	5 internal 24 external
Packages	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN
Timers (16-/8-bit)	4/3	
Serial Communications	2 UART, 2 I^2C, 1 SPI	
Enhanced Complementary Waveform Generator (ECWG)	3	
Zero-Cross Detect (ZCD)	1	
Data Signal Modulator (DSM)	1	
Signal Measurement Timer (SMT)	1	
5-Bit Digital-to-Analog Converter (DAC)	1	
Numerically Controlled Oscillator (NCO)	1	
Comparator Module	2	
Direct Memory Access (DMA)	2	
Configurable Logic Cell (CLC)	4	
Peripheral Pin Select (PPS)	Yes	
Peripheral Module Disable (PMD)	Yes	
16-bit CRC with Scanner	Yes	
Programmable High/Low-Voltage Detect (HLVD)	Yes	
Programmable Brown-out Reset (BOR)	Yes	
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Overflow, Stack Underflow (PWRT, OST), $\overline{MCLR}$, WDT, Memory Execution Violation	
Instruction Set	81 Instructions; 87 with Extended Instruction Set enabled	
Operating Frequency	64 MHz	

Note 1: PORTE contains the single RE3 input-only pin.

TABLE 4-11: REGISTER FILE SUMMARY FOR PIC18(L)F24/25K42 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
3FCEh	PORTE	—	—	—	—	RE3	—	—	—	----x---
3FCDh	—	Unimplemented								—
3FCCh	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxxxxxx
3FCBh	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxxxxxx
3FCAh	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxxxxxx
3FC9h - 3FC5h	—	Unimplemented								—
3FC4h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	11111111
3FC3h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	11111111
3FC2h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	11111111
3FC1h - 3FBDh	—	Unimplemented								—
3FBCh	LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxxxxxx
3FBBh	LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxxxxxx
3FBAh	LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxxxxxx
3FB9h	T0CON1	CS<2:0>			ASYNC	CKPS<3:0>				00000000
3FB8h	T0CON0	EN	—	OUT	MD16	OUTPS				0-000000
3FB7h	TMR0H	TMR0H								11111111
3FB6h	TMR0L	TMR0L								00000000
3FB5h	T1CLK	CS								---00000
3FB4h	T1GATE	GSS								---00000
3FB3h	T1GCON	GE	GPOL	GTM	GSPM	GGO	GVAL	—	—	00000x--
3FB2h	T1CON	—	—	CKPS<1:0>		—	$\overline{\text{SYNC}}$	RD16	ON	--00-000
3FB1h	TMR1H	TMR1H								00000000
3FB0h	TMR1L	TMR1L								00000000
3FAFh	T2RST	—	—	—	RSEL					---00000
3FAEh	T2CLK	—	—	—	—	CS				----0000
3FADh	T2HLT	PSYNC	CKPOL	CKSYNC	MODE					00000000
3FACh	T2CON	ON	CKPS			OUTPS				00000000
3FABh	T2PR	PR2								11111111
3FAAh	T2TMR	TMR2								00000000
3FA9h	T3CLK	CS								---00000
3FA8h	T3GATE	GSS								---00000
3FA7h	T3GCON	GE	GPOL	GTM	GSPM	GGO	GVAL	—	—	00000x--
3FA6h	T3CON	—	—	CKPS		—	NOT_SYNC	RD16	ON	--00-000
3FA5h	TMR3H	TMR3H								00000000
3FA4h	TMR3L	TMR3L								00000000
3FA3h	T4RST	—	—	—	RSEL					---00000
3FA2h	T4CLK	—	—	—	—	CS				----0000
3FA1h	T4HLT	PSYNC	CKPOL	CKSYNC	MODE					00000000
3FA0h	T4CON	ON	CKPS			OUTPS				00000000
3F9Fh	T4PR	PR4								11111111
3F9Eh	T4TMR	TMR4								00000000
3F9Dh	T5CLK	CS								---00000
3F9Ch	T5GATE	GSS								---00000
3F9Bh	T5GCON	GE	GPOL	GTM	GSPM	GGO	GVAL	—	—	00000x--
3F9Ah	T5CON	—	—	CKPS		—	NOT_SYNC	RD16	ON	--00-000
3F99h	TMR5H	TMR5H								00000000
3F98h	TMR5L	TMR5L								00000000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not present in LF devices.

REGISTER 5-7: CONFIGURATION WORD 4L (30 0006h)

R/W-1	U-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{WRTAPP}}$ (1)	—	—	$\overline{\text{SAFEN}}$ (2)	$\overline{\text{BBEN}}$ (2)	BBSIZE<2:0> (3)		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘1’	
-n = Value for blank device	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{WRTAPP}}$:** Application Block write protection bit
1 = Application Block NOT write protected
0 = Application Block write protected

bit 6-5 **Unimplemented:** Read as ‘1’

bit 4 **$\overline{\text{SAFEN}}$:** Storage Area Flash Enable bit
1 = SAF disabled
0 = SAF enabled

bit 3 **$\overline{\text{BBEN}}$:** Boot Block Enable bit
1 = Boot Block disabled
0 = Boot Block enabled

bit 2-0 **BBSIZE<2:0>:** Boot Block Size Selection bits
Refer to Table 5-1.

Note 1: Once protection is enabled through ICSP or a self write, it can only be reset through a bulk erase.
2: See Table 4-2 for Program Flash Memory Partition.
3: BBSIZE bits can only be changed when $\overline{\text{BBEN}}$ = 1. Once $\overline{\text{BBEN}}$ = 0, BBSIZE can only be changed through a bulk erase.

TABLE 5-1: BOOT BLOCK SIZE BITS

$\overline{\text{BBEN}}$	BBSIZE<2:0>	Boot Block Size (words)	END_ADDRESS_BOOT	Device Size(1)	
				8K Words	16K Words
1	xxx	0	—	X	X
0	111	512	00 03FFh	X	X
0	110	1024	00 07FFh	X	X
0	101	2048	00 0FFFh	X	X
0	100	4096	00 1FFFh	X	X
0	011	8192	00 3FFFh	—	X
0	010	16384	00 7FFFh	—	—
0	001	32768	00 FFFFh	Note 2	
0	000	32768	00 FFFFh	—	—

Note 1: See Table 5-1 for Device Size.
2: The maximum boot block size is half the user program memory size. All selections higher than the maximum size default to maximum boot block size of half PFM. For example, all settings of BBSIZE = 000 through BBSIZE = 100, default to a boot block size of 4 kW on a 8 kW device.

REGISTER 8-3: PCON1: POWER CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W/HC-1/u	U-0
—	—	—	—	—	—	$\overline{\text{MEMV}}$	—
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	-m/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **$\overline{\text{MEMV}}$:** Memory Violation Flag bit
1 = No memory violation Reset occurred or set to ‘1’ by firmware
0 = A memory violation Reset occurred (set to ‘0’ in hardware when a Memory Violation occurs)

bit 0 **Unimplemented:** Read as ‘0’

TABLE 8-4: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	90
PCON0	STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	95
PCON1	—	—	—	—	—	—	$\overline{\text{MEMV}}$	—	96

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by Resets.

REGISTER 11-19: PIE5: PERIPHERAL INTERRUPT ENABLE REGISTER 5

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
I2C2TXIE	I2C2RXIE	DMA2AIE	DMA2ORIE	DMA2DCNTIE	DMA2SCNTIE	C2IE	INT1IE
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 **I2C2TXIE:** I^2C2 Transmit Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **I2C2RXIE:** I^2C2 Receive Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **DMA2AIE:** DMA2 Abort Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **DMA2ORIE:** DMA2 Overrun Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **DMA2DCNTIE:** DMA2 Destination Count Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **DMA2SCNTIE:** DMA2 Source Count Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **C2IE:** C2 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **INT1IE:** External Interrupt 1 Enable bit
1 = Enabled
0 = Disabled

14.0 8x8 HARDWARE MULTIPLIER

14.1 Introduction

All PIC18 devices include an 8x8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 14-1.

14.2 Operation

Example 14-1 shows the instruction sequence for an 8x8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 14-2 shows the sequence to do an 8x8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 14-1: 8x8 UNSIGNED MULTIPLY ROUTINE

```
MOVF   ARG1, W    ;
MULWF  ARG2       ; ARG1 * ARG2 ->
                  ; PRODH:PRODL
```

EXAMPLE 14-2: 8x8 SIGNED MULTIPLY ROUTINE

```
MOVF   ARG1, W
MULWF  ARG2       ; ARG1 * ARG2 ->
                  ; PRODH:PRODL
BTFSC  ARG2, SB   ; Test Sign Bit
SUBWF  PRODH, F   ; PRODH = PRODH
                  ;          - ARG1
MOVF   ARG2, W
BTFSC  ARG1, SB   ; Test Sign Bit
SUBWF  PRODH, F   ; PRODH = PRODH
                  ;          - ARG2
```

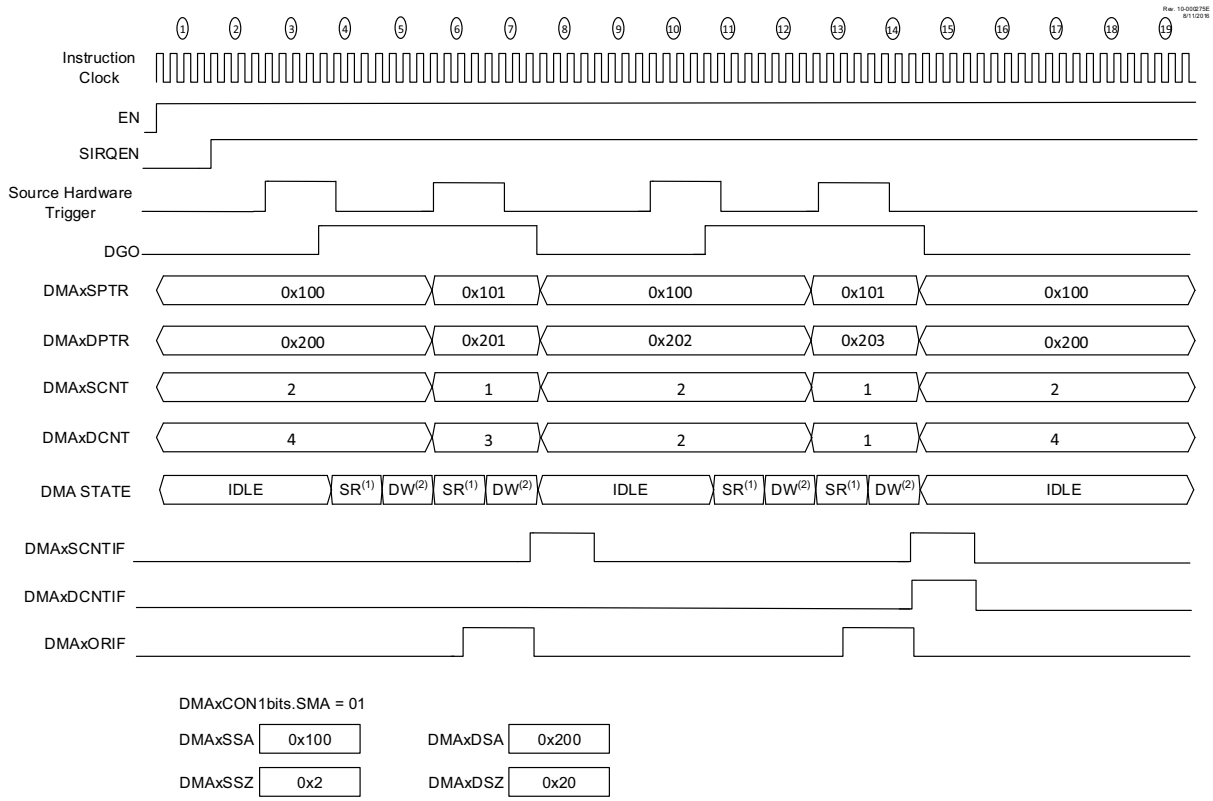
TABLE 14-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time			
				@ 64 MHz	@ 40 MHz	@ 10 MHz	@ 4 MHz
8x8 unsigned	Without hardware multiply	13	69	4.3 μs	6.9 μs	27.6 μs	69 μs
	Hardware multiply	1	1	62.5 ns	100 ns	400 ns	1 μs
8x8 signed	Without hardware multiply	33	91	5.7 μs	9.1 μs	36.4 μs	91 μs
	Hardware multiply	6	6	375 ns	600 ns	2.4 μs	6 μs
16x16 unsigned	Without hardware multiply	21	242	15.1 μs	24.2 μs	96.8 μs	242 μs
	Hardware multiply	28	28	1.8 μs	2.8 μs	11.2 μs	28 μs
16x16 signed	Without hardware multiply	52	254	15.9 μs	25.4 μs	102.6 μs	254 μs
	Hardware multiply	35	40	2.5 μs	4.0 μs	16.0 μs	40 μs

17.9.5 OVERRUN INTERRUPT

The Overrun Interrupt flag is set if the DMA receives a trigger to start a new message before the current message is completed.

FIGURE 17-9: OVERRUN INTERRUPT



Note 1: SR - Source Read

2: DW - Destination Write

REGISTER 17-15: DMAxDSAH – DMAx DESTINATION START ADDRESS HIGH 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
DSA<15:8>							
bit 7							bit 0

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

bit 7-0 **DSA<15:8>:** Destination Start Address bits

REGISTER 17-16: DMAxDPTRL – DMAx DESTINATION POINTER LOW REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DPTR<7:0>							
bit 7							bit 0

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

bit 7-0 **DPTR<7:0>:** Current Destination Address Pointer

REGISTER 17-17: DMAxDPTRH – DMAx DESTINATION POINTER HIGH REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DPTR<15:8>							
bit 7							bit 0

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

bit 7-0 **DPTR<15:8>:** Current Destination Address Pointer

TABLE 17-2: DMAxSIRQ AND DMAxAIRQ INTERRUPT SOURCES

DMAxSIRQ DMAxAIRQ	Trigger Source	Level Triggered	DMAxSIRQ DMAxAIRQ	Trigger Source	Level Triggered
0	Reserved		42	DMA2SCNT	No
1	LVD	No	43	DMA2DCNT	No
2	OSF	No	44	DMA2OR	No
3	CSW	No	45	DMA2A	No
4	NVM	No	46	I2C2RX	Yes
5	SCAN	No	47	I2C2TX	Yes
6	CRC	No	48	I2C2	Yes
7	IOC	Yes	49	I2C2E	Yes
8	INT0	No	50	U2RX	Yes
9	ZCD	No	51	U2TX	Yes
10	AD	No	52	U2E	Yes
11	ADT	No	53	U2	No
12	CMP1	No	54	TMR3	No
13	SMT1	No	55	TMR3G	No
14	SMT1PRA	No	56	TMR4	No
15	SMT1PWA	No	57	CCP2	No
16	DMA1SCNT	No	58	Reserved	
17	DMA1DCNT	No	59	CWG2	No
18	DMA1OR	No	60	CLC2	No
19	DMA1A	No	61	INT2	No
20	SPI1RX	Yes	62	Reserved	
21	SPI1TX	Yes	63	Reserved	
22	SPI1	Yes	64	Reserved	
23	I2C1RX	Yes	65	Reserved	
24	I2C1TX	Yes	66	Reserved	
25	I2C1	Yes	67	Reserved	
26	I2C1E	Yes	68	Reserved	
27	U1RX	Yes	69	Reserved	
28	U1TX	Yes	70	TMR5	No
29	U1E	Yes	71	TMR5G	No
30	U1	No	72	TMR6	No
31	TMR0	No	73	CCP3	No
32	TMR1	No	74	CWG3	No
33	TMR1G	No	75	CLC3	No
34	TMR2	No	76	Reserved	
35	CCP1	No	77	Reserved	
36	Reserved		78	Reserved	
37	NCO	No	79	Reserved	
38	CWG1	No	80	CCP4	No
39	CLC1	No	81	CLC4	No
40	INT1	No	82 – 127	Reserved	
41	CMP2	No			

Note 1: All trigger sources that are not Level-triggered are Edge-triggered.

TABLE 17-3: SUMMARY OF REGISTERS ASSOCIATED WITH DMA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
DMAxCON0	EN	SIRQEN	DGO	—	—	AIRQEN	—	XIP	252
DMAxCON1	DMODE<1:0>		DSTP	SMR<1:0>		SMODE<1:0>		SSTP	253
DMAxBUF	DBUF7	DBUF6	DBUF5	DBUF4	DBUF3	DBUF2	DBUF1	DBUF0	254
DMAxSSAL	SSA<7:0>								254
DMAxSSAH	SSA<15:8>								254
DMAxSSAU	—	—	SSA<21:16>						255
DMAxSPTRL	SPTR<7:0>								255
DMAxSPTRH	SPTR<15:8>								255
DMAxSPTRU	—	—	SPTR<21:16>						256
DMAxSSZL	SSZ<7:0>								256
DMAxSSZH	—	—	—	—	SSZ<11:8>				256
DMAxSCNTL	SCNT<7:0>								257
DMAxSCNTH	—	—	—	—	SCNT<11:8>				257
DMAxDSAL	DSA<7:0>								257
DMAxDSAH	DSA<15:8>								258
DMAxDPTRL	DPTR<7:0>								258
DMAxDPTRH	DPTR<15:8>								258
DMAxDSZL	DSZ<7:0>								259
DMAxDSZH	—	—	—	—	DSZ<11:8>				259
DMAxDCNTL	DCNT<7:0>								259
DMAxDCNTH	—	—	—	—	DCNT<11:8>				260
DMAxSIRQ	—	SIRQ6	SIRQ5	SIRQ4	SIRQ3	SIRQ2	SIRQ1	SIRQ0	260
DMAxAIRQ	—	AIRQ6	AIRQ5	AIRQ4	AIRQ3	AIRQ2	AIRQ1	AIRQ0	260

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by DMA.

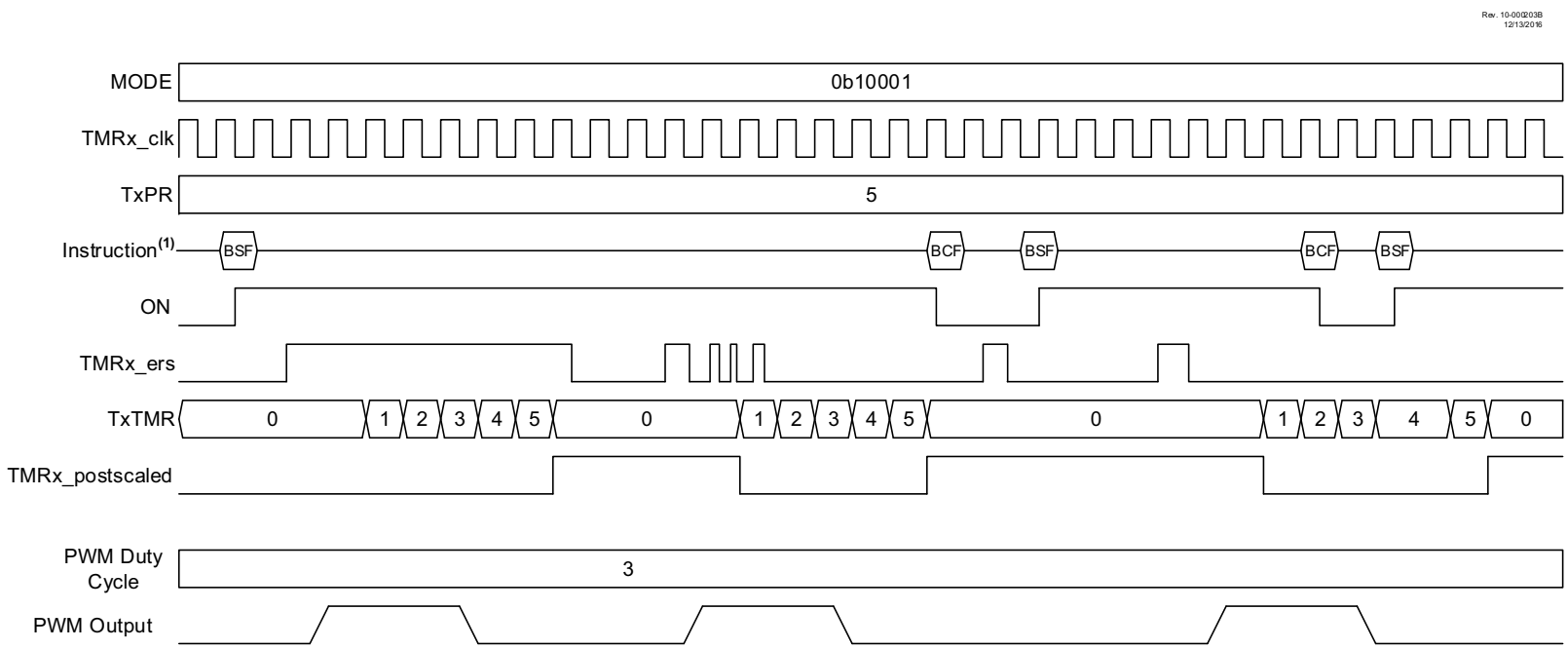
24.5.9 EDGE-TRIGGERED MONOSTABLE MODES

The Edge-Triggered Monostable modes start the timer on an edge from the external Reset signal input, after the ON bit is set, and stop incrementing the timer when the timer matches the T2PR period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = `10001`)
- Falling edge (MODE<4:0> = `10010`)
- Rising or Falling edge (MODE<4:0> = `10011`)

When an Edge-Triggered Monostable mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external Reset signal edge that starts the timer, but will not go active when the timer matches the T2PR value. While the timer is incrementing, additional edges on the external Reset signal will not affect the CCP PWM.

FIGURE 24-12: RISING EDGE-TRIGGERED MONOSTABLE MODE TIMING DIAGRAM (MODE = `10001`)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

REGISTER 29-11: CLCDATA: CLC DATA OUTPUT

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	CLC4OUT	CLC3OUT	CLC2OUT	CLC1OUT
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as ‘0’
bit 3 **CLC4OUT:** Mirror copy of OUT bit of CLC4CON register
bit 2 **CLC3OUT:** Mirror copy of OUT bit of CLC3CON register
bit 1 **CLC2OUT:** Mirror copy of OUT bit of CLC2CON register
bit 0 **CLC1OUT:** Mirror copy of OUT bit of CLC1CON register

TABLE 29-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CLCxCON	EN	—	OUT	INTP	INTN	MODE<2:0>			459
CLCxPOL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	460
CLCxSEL0	—	—	D1S<5:0>						461
CLCxSEL1	—	—	D2S<5:0>						461
CLCxSEL2	—	—	D3S<5:0>						461
CLCxSEL3	—	—	D4S<5:0>						461
CLCxGLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	462
CLCxGLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	463
CLCxGLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	464
CLCxGLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	465
CLCDATA	—	—	—	—	CLC4OUT	CLC3OUT	CLC2OUT	CLC1OUT	465

Legend: — = unimplemented, read as ‘0’. Shaded cells are unused by the CLCx modules.

33.6 DALI Mode (UART1 only)

DALI is a protocol used to control lighting in large buildings such as offices and factories. It consists of two modes: 'Control Device' and 'Control Gear'. A control device is the main controller that sends out commands to the lighting fixtures. The lighting fixture itself is known as control gear. All bit transmission is done in Manchester encoding, which is done by the hardware.

Manchester encoding contains the clock and data in a single bit stream. A bit always has a transition in the middle of the bit period and may or may not have a transition at the bit period boundaries. When consecutive bits are the same value then there will be a transition at the bit boundary. When the bit value changes then there will not be a transition at the bit boundary.

When ABDEN = 0, the receiver bit rate is determined by the BRG register. Otherwise, when ABDEN = 1, the first bit synchronizes the receiver to the transmitter and sets the receiver bit rate. The Start bit low period is measured and used as the timing reference for all data bits. The ABDOVF bit will be set if the Start bit low period causes the measurement counter to overflow. All bits that follow the Start bit are data bits. The bit stream terminates when there is no transition sensed in the middle of a bit period. See Figure 33-7.

The DALI wire is half-duplex: The transmit and receive lines are electrically tied together through an interface circuit such as a diode bridge. Wait periods between frames ensure that the Forward and Back Frames do not collide.

Unlike all other protocols, DALI is transmitted MSb first. The transaction starts when the 'control device' starts a transmission. A control device transmission is called the 'Forward Frame' and consists of two bytes in DALI 1.0 or three bytes in DALI 2.0. The first byte is the control followed by one or two data bytes. Typical frame timing is shown in Figure 33-8.

When writing code for DALI 2.0 Devices, where three bytes must be transmitted, the software must write the third byte to UxTXB as soon as UxTXIF goes true and before the output shifter becomes empty. This will ensure that three bytes are transmitted back-to-back without interruption.

All control gear on the line receive the 'Forward Frame'. One of the control gear may respond to this with a single byte in reply, called the 'Back Frame'. The DALI protocol requires that the Back Frame must begin to be received between 3.5 to 11 bit periods after the Forward Frame. If a Back Frame is received, the control device is required to wait a minimum of 11 bit periods after the end of the Back Frame. After this time, the control device is free to transmit another Forward Frame. See Figure 33-9.

Forward and Back Frames are terminated by two idle bit periods or Stop bits. Normally these start in the first bit period of a byte. If both Stop bits are valid, the byte reception is terminated without further action.

If either Stop bit is invalid, the frame is tagged as invalid by saving a null byte, with the framing error set, in the receive FIFO.

A framing error will also occur when a no-transition is detected in the middle of a bit period and the byte in progress is not complete. In this case, the byte will be saved with the FERIF bit set.

33.6.1 DALI DEVICE TRANSMITTER

DALI Control Device mode is configured with the following settings:

- MODE<3:0> = 1000
- TXEN = 1
- RXEN = 1
- UxP1 = Forward Frames are held for transmission this number of half-bit periods after the completion of a Forward or Back Frame.
- UxP2 = Forward/Back Frame threshold delimiter. Any reception that starts this number of half bit periods after the completion of a Forward or Back Frame is detected as Forward Frame and sets the PERIF flag of the corresponding received byte.
- UxBRGH:L = Value to achieve 1200 baud rate
- TXPOL = appropriate polarity for interface circuit
- STP<1:0> = 10 for two Stop bits
- RxyPPS = TX pin selection code
- TX pin TRIS control = 0
- ON = 1

A Forward Frame is initiated by writing the control byte to the UxTXB register. Each data byte after the control byte must be written to the UxTXB register as soon as UxTXIF goes true. It is important that every write is performed after UxTXIF goes true, to ensure the transmit buffer is ready to accept the byte. Each write must also occur before the TXMTIF bit goes true, to ensure the Forward Frame bit stream is generated without interruption.

When TXMTIF goes true, indicating the transmit shift register has completed sending the last byte in the frame, the TX output is held in the Idle state for the number of half-bit periods selected by the STP<1:0> bits in the UxCON2 register.

After the last Stop bit, the TX output is held in the Idle state for an additional wait time determined by the half-bit period count in the UxP1 register. An 11 bit period delay requires a value of 22 in UxP1L.

Any writes to the UxTXB register that occur after TXMTIF goes true, but before the UxP1 wait time, will be held and then transmitted immediately following the wait time. If a Back Frame is received during the wait time, any bytes that may have been written to UxTXB will not be transmitted until after the Back Frame plus the UxP1 wait time.

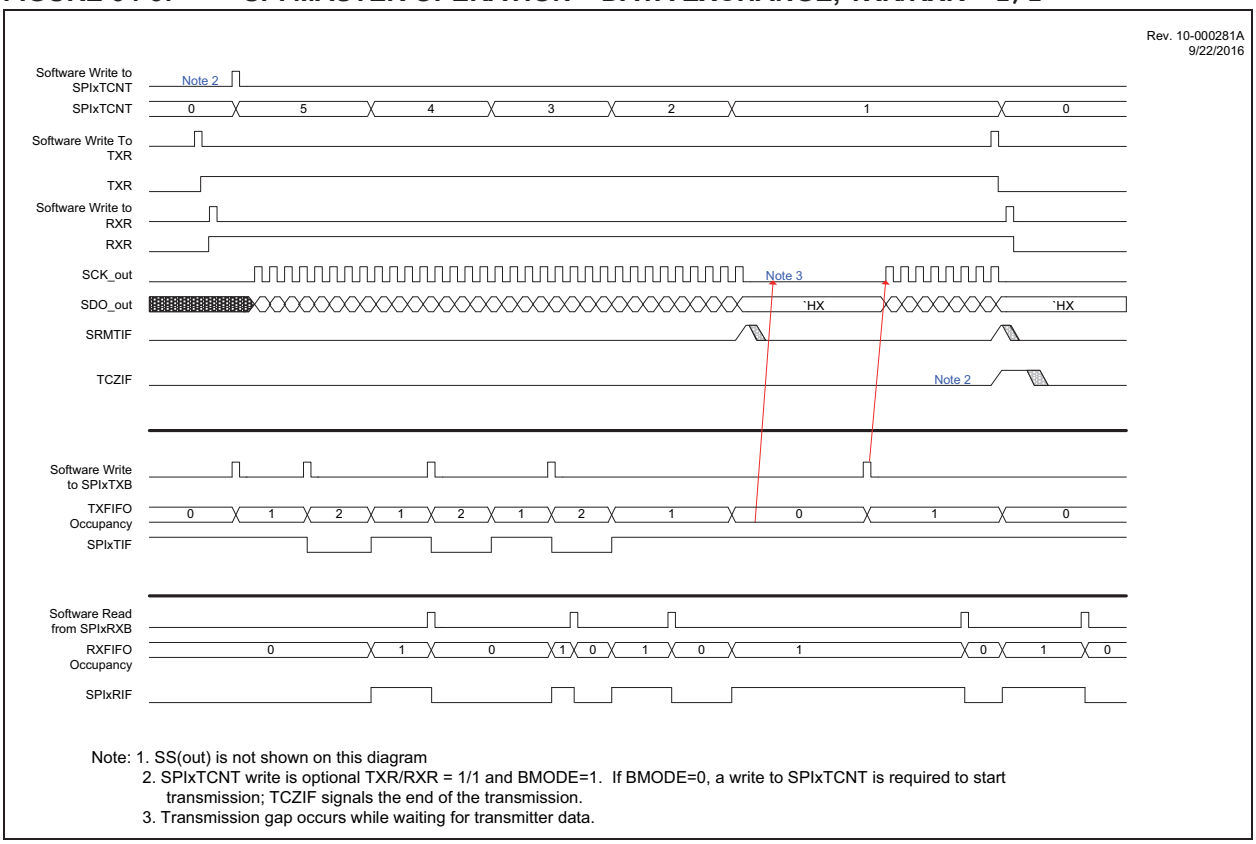
34.5.1 FULL DUPLEX MODE

When both TXR and RXR are set, the SPI master is in Full Duplex mode. In this mode, data transfer triggering is affected by the BMODE bit of SPIxCON0.

When BMODE = 1, data transfers will occur whenever both the RXFIFO is not full and there is data present in the TXFIFO. In practice, as long as the RXFIFO is not full, data will be transmitted/received as soon as the SPIxTxB register is written to, matching functionality of SPI (MSSP) modules on older 8-bit Microchip devices. The SPIxTCNT will decrement with each transfer. However, when SPIxTCNT is zero the next transfer is not inhibited and the corresponding SPIxTCNT decrement will cause the count to roll over to the maximum value. Figure 34-3 shows an example of a communication using this mode.

When BMODE = 0, the transfer counter (SPIxTCNTH/SPIxTCNTL) must also be written to before transfers will occur, and transfers will cease when the transfer counter reaches '0'. For example, if SPIxTXB is written twice and then SPIxTCTL is written with '3' then the transfer will start with the SPIxTCTL write. The two bytes in the TXFIFO will be sent after which the transfer will suspend until the third and last byte is written to SPIxTXB.

FIGURE 34-3: SPI MASTER OPERATION – DATA EXCHANGE, TXR/RXR = 1/1

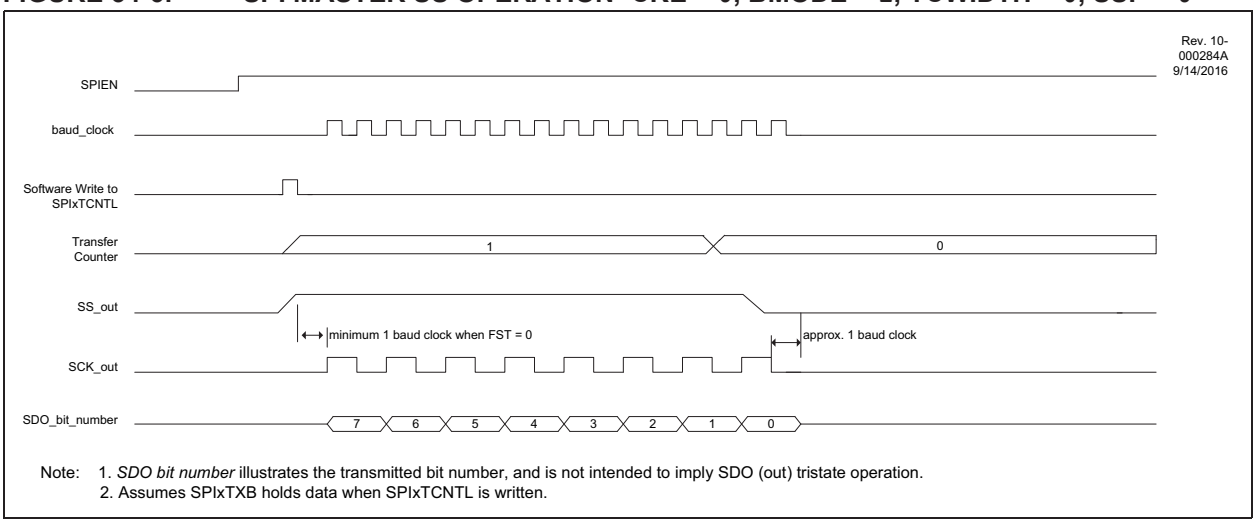


34.5.5 MASTER MODE SLAVE SELECT CONTROL

34.5.5.1 Hardware Slave Select Control

This SPI module allows for direct hardware control of a Slave Select output. The Slave Select output SS(out) is controlled both directly, through the SSET bit of SPIxCON2, as well indirectly by the hardware while the transfer counter is non-zero (see **Section 34.4 "Transfer Counter"**). SS(out) is steered by the PPS registers to pins (see **Section 19.2 "PPS Outputs"**) and its polarity is controlled by the SSP bit of SPIxCON1. Setting the SSET bit will also assert SS(out). Clearing the SSET bit will leave SS(out) to be controlled by the Transfer Counter. When the Transfer Counter is loaded, the SPI module will automatically assert the SS. When the Transfer Counter decrements to zero, the SPI module will deassert SS either one baud period after the final SCK pulse of the final transfer (if CKE/SMP = `0/1`) or one half baud period otherwise (see Figure 34-6).

FIGURE 34-6: SPI MASTER SS OPERATION- CKE = `0`, BMODE = `1`, TCWIDTH = `0`, SSP = `0`



Note: 1. *SDO bit number* illustrates the transmitted bit number, and is not intended to imply SDO (out) tristate operation.
2. Assumes SPIxTXB holds data when SPIxTCNTL is written.

34.5.5.2 Software Slave Select Control

Slave Select can also be controlled through software via a general purpose I/O pin. In this case, ensure that the pin in question is configured as a GPIO through PPS (see **Section 19.2 "PPS Outputs"**), and ensure that the pin is set as an output (clear the appropriate bit in the appropriate TRIS register). In this case, SSET will not affect the slave select, the Transfer Counter will not automatically control the slave select output, and all setting and clearing of the slave select output line must be directly controlled by software.

35.5.10 MASTER RECEPTION IN 7-BIT ADDRESSING MODE

This section describes the sequence of events for the I^2C module configured as an I^2C master in 7-bit Addressing mode and is receiving data. Figure 35-20 is used as a visual reference for this description.

1. Master software loads slave address in I2CxADB1 with R/W bit = d and number of bytes to be received in one sequence in I2CxCNT register.
2. Master hardware waits for BFRE bit to be set; then shifts out start and address with R/W = 1.
3. Master sends out the 9th SCL pulse for ACK, master hardware clocks in ACK from Slave
4. If ABD = 0; i.e. Address buffers are enabled

If NACK, master hardware sends Stop or sets MDR (if RSEN = 1) and waits for user software to write to S bit for restart.

If ABD = 1; i.e. Address buffers are disabled

If NACK, master hardware sends Stop or sets MDR (if RSEN = 1) and waits for user software to load the new address into I2CxTXB. Software writes to the S bit are ignored in this case.

5. If ACK, master hardware receives 7-bits of data into the shift register.
6. If the receive buffer is full (i.e. RXBF = 1), clock is stretched on 7th falling SCL edge.
7. Master software must read previous data out of I2CxRXB to clear RXBF.
8. Master hardware receives 8th bit of data into the shift register and loads it into I2CxRXB, sets I2CxRXIF and RXBF bits. I2CxCNT is decremented.
9. If I2CxCNT! = 0, master hardware clocks out ACKDT as ACK value to slave. If I2CxCNT = 0, master hardware clocks out ACKCNT as ACK value to slave. It is up to the user to set the values of ACKDT and ACKCNT correctly. If the user does not set ACKCNT to '1', the master hardware will never send a NACK when I2CxCNT becomes zero. Since a NACK was not seen on the bus, the master hardware will also not assert a Stop condition.
10. Go to step 4.

REGISTER 35-3: I2CxCON2 – I²C CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACNT	GCEN	FME	ADB	SDAHT<1:0>		BFRET<1:0>	
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	HS = Hardware set	HC = Hardware clear

bit 7 **ACNT:** Auto-Load I²C Count Register Enable bit
1 = The first received or transmitted byte after the address, is automatically loaded into the I2CCNT register. The I2CCNT register is loaded at the same time as the value is moved to/from the shifter. ACKDT is used to determine the ACK/NACK value for the address bytes and first data byte of a received message. This prevents a I2CCNT<NACK> from being sent for the byte that would update the I2CCNT register.
0 = Auto-load of I2CCNT disabled

bit 6 **GCEN:** General Call Address Enable bit (MODE<2:0> = 00x & 11x)
1 = General call address, 0x00, causes address match event
0 = General call address disabled

bit 5 **FME:** Fast Mode Enable bit
1 = SCL is sampled high only once before driving SCL low. ($F_{SCL} = F_{CLK}/4$)
0 = SCL is sampled high twice before driving SCL low. ($F_{SCL} = F_{CLK}/5$)

bit 4 **ADB:** Address Data Buffer Disable bit
1 = Received address data is loaded into both the I2CADB and I2CRXB
Transmitted address data is loaded from the I2CTXB
0 = Received address data is loaded only into the I2CADB
Transmitted address data is loaded from the I2CADB0/1 registers.

bit 3-2 **SDAHT<1:0>:** SDA Hold Time Selection bits
11 = Reserved
10 = Minimum of 30 ns hold time on SDA after the falling edge of SCL
01 = Minimum of 100 ns hold time on SDA after the falling edge of SCL
00 = Minimum of 300 ns hold time on SDA after the falling edge of SCL

bit 1-0 **BFRET<1:0>:** Bus Free Time Selection bits
11 = 64 I²C Clock pulses
10 = 32 I²C Clock pulses
01 = 16 I²C Clock pulses
00 = 8 I²C Clock pulses

40.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 40-1) contains Control and Status bits for the following:

- Enable
- Output
- Output polarity
- Hysteresis enable
- Timer1 output synchronization

The CMxCON1 register (see Register 40-2) contains Control bits for the following:

- Interrupt on positive/negative edge enables

The CMxPCH and CMxNCH registers are used to select the positive and negative input channels, respectively.

40.2.1 COMPARATOR ENABLE

Setting the EN bit of the CMxCON0 register enables the comparator for operation. Clearing the EN bit disables the comparator resulting in minimum current consumption.

40.2.2 COMPARATOR OUTPUT

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the CxOUT bit of the CMOUT register.

The comparator output can also be routed to an external pin through the RxyPPS register (Register 19-2). The corresponding TRIS bit must be clear to enable the pin as an output.

Note 1:	The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

40.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the POL bit of the CMxCON0 register. Clearing the POL bit results in a non-inverted output.

Table 40-1 shows the output state versus input conditions, including polarity control.

TABLE 40-1: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	POL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

RCALL	Relative Call
Syntax:	RCALL n
Operands:	-1024 ≤ n ≤ 1023
Operation:	(PC) + 2 → TOS, (PC) + 2 + 2n → PC
Status Affected:	None
Encoding:	1101 | 1nnn | nnnn | nnnn
Description:	Subroutine call with a jump up to 1K from the current location. First, return address (PC + 2) is pushed onto the stack. Then, add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is a 2-cycle instruction.
Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'n' PUSH PC to stack	Process Data	Write to PC
No operation	No operation	No operation	No operation

Example: `HERE RCALL Jump`

Before Instruction
PC = Address (HERE)
After Instruction
PC = Address (Jump)
TOS = Address (HERE + 2)

RESET	Reset
Syntax:	RESET
Operands:	None
Operation:	Reset all registers and flags that are affected by a $\overline{\text{MCLR}}$ Reset.
Status Affected:	All
Encoding:	0000 | 0000 | 1111 | 1111
Description:	This instruction provides a way to execute a $\overline{\text{MCLR}}$ Reset by software.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Start Reset	No operation	No operation

Example: `RESET`

After Instruction
Registers = Reset Value
Flags* = Reset Value

RETURN — Return from Subroutine

Syntax:	RETURN {s}
Operands:	s ∈ [0,1]
Operation:	(TOS) → PC, if s = 1 (WS) → W, (STATUSS) → Status, (BSRS) → BSR, PCLATU, PCLATH are unchanged
Status Affected:	None
Encoding:	0000 | 0000 | 0001 | 001s
Description:	Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers, WS, STATUSS and BSRS, are loaded into their corresponding registers, W, Status and BSR. If 's' = 0, no update of these registers occurs (default).
Words:	1
Cycles:	2

Q Cycle Activity:

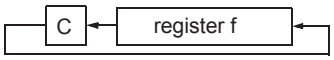
Q1	Q2	Q3	Q4
Decode	No operation	Process Data	POP PC from stack
No operation	No operation	No operation	No operation

Example: `RETURN`

After Instruction:
PC = TOS

RLCF — Rotate Left f through Carry

Syntax:	RLCF f {,d {,a}}
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	(f<n>) → dest<n + 1>, (f<7>) → C, (C) → dest<0>
Status Affected:	C, N, Z
Encoding:	0011 | 01da | ffff | ffff
Description:	The contents of register 'f' are rotated one bit to the left through the CARRY flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 43.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `RLCF REG, 0, 0`

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1