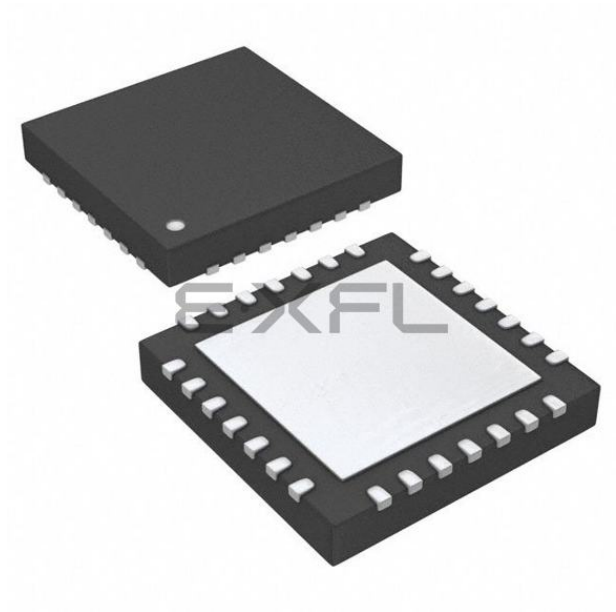


Welcome to E-XFL.COM

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, POR, PWM, WDT
Number of I/O	44
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 43x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f15385-i-mv

TABLE 1-2: PIC16(L)F15356 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/ANA0/C1IN0-/C2IN0-/CLCIN0(1)/IOCA0	RA0	TTL/ST	CMOS/OD	General purpose I/O.
	ANA0	AN	—	ADC Channel A0 input.
	C1IN0-	AN	—	Comparator 1 negative input.
	C2IN0-	AN	—	Comparator 2 negative input.
	CLCIN0(1)	TTL/ST	—	Configurable Logic Cell source input.
	IOCA0	TTL/ST	—	Interrupt-on-change input.
RA1/ANA1/C1IN1-/C2IN1-/CLCIN1(1)/IOCA1	RA1	TTL/ST	CMOS/OD	General purpose I/O.
	ANA1	AN	—	ADC Channel A1 input.
	C1IN1-	AN	—	Comparator 1 negative input.
	C2IN1-	AN	—	Comparator 2 negative input.
	CLCIN1(1)	TTL/ST	—	Configurable Logic Cell source input.
	IOCA1	TTL/ST	—	Interrupt-on-change input.
RA2/ANA2/C1IN0+/C2IN0+/DAC1OUT1/IOCA2	RA2	TTL/ST	CMOS/OD	General purpose I/O.
	ANA2	AN	—	ADC Channel A2 input.
	C1IN0+	AN	—	Comparator 2 positive input.
	C2IN0+	AN	—	Comparator 2 positive input.
	DAC1OUT1	—	AN	Digital-to-Analog Converter output.
	IOCA2	TTL/ST	—	Interrupt-on-change input.
RA3/ANA3/C1IN1+/VREF+/IOCA3/DAC1REF+	RA3	TTL/ST	CMOS/OD	General purpose I/O.
	ANA3	AN	—	ADC Channel A3 input.
	C1IN1+	AN	—	Comparator 1 positive input.
	VREF+	AN	—	External ADC and/or DAC positive reference input.
	IOCA3	TTL/ST	—	Interrupt-on-change input.
	DAC1REF+	TTL/ST	AN	DAC positive reference.
RA4/ANA4/T0CKI(1)/IOCA4	RA4	TTL/ST	CMOS/OD	General purpose I/O.
	ANA4	AN	—	ADC Channel A4 input.
	T0CKI(1)	TTL/ST	—	Timer0 clock input.
	IOCA4	TTL/ST	—	Interrupt-on-change input.
RA5/ANA5/$\overline{SS1}$(1)/IOCA5	RA5	TTL/ST	CMOS/OD	General purpose I/O.
	ANA5	AN	—	ADC Channel A5 input.
	$\overline{SS1}$(1)	TTL/ST	—	MSSP1 SPI slave select input.
	IOCA5	TTL/ST	—	Interrupt-on-change input.

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

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 15-4 for details on which PORT pins may be used for this signal.

2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 15-3.

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

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

TABLE 4-4: PIC16(L)F15356/75/76/85/86 MEMORY MAP, BANKS 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	Core Register (Table 4-3)	080h	Core Register (Table 4-3)	100h	Core Register (Table 4-3)	180h	Core Register (Table 4-3)	200h	Core Register (Table 4-3)	280h	Core Register (Table 4-3)	300h	Core Register (Table 4-3)	380h	Core Register (Table 4-3)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	—	10Ch	—	18Ch	SSP1BUF	20Ch	TMR1L	28Ch	TMR2	30Ch	CCPR1L	38Ch	PWM6DCL
00Dh	PORTB	08Dh	—	10Dh	—	18Dh	SSP1ADD	20Dh	TMR1H	28Dh	PR2	30Dh	CCPR1H	38Dh	PWM6DCH
00Eh	PORTC	08Eh	—	10Eh	—	18Eh	SSP1MASK	20Eh	T1CON	28Eh	T2CON	30Eh	CCP1CON	38Eh	PWM6CON
00Fh	PORTD(2)	08Fh	—	10Fh	—	18Fh	SSP1STAT	20Fh	T1GCON	28Fh	T2HLT	30Fh	CCP1CAP	38Fh	—
010h	PORTE	090h	—	110h	—	190h	SSP1CON1	210h	T1GATE	290h	T2CLK	310h	CCPR2L	390h	—
011h	PORTF(3)	091h	—	111h	—	191h	SSP1CON2	211h	T1CLK	291h	T2ERS	311h	CCPR2H	391h	—
012h	TRISA	092h	—	112h	—	192h	SSP1CON3	212h	—	292h	—	312h	CCP2CON	392h	—
013h	TRISB	093h	—	113h	—	193h	—	213h	—	293h	—	313h	CCP2CAP	393h	—
014h	TRISC	094h	—	114h	—	194h	—	214h	—	294h	—	314h	PWM3DCL	394h	—
015h	TRISD(2)	095h	—	115h	—	195h	—	215h	—	295h	—	315h	PWM3DCH	395h	—
016h	TRISE	096h	—	116h	—	196h	SSP2BUF	216h	—	296h	—	316h	PWM3CON	396h	—
017h	TRISF(3)	097h	—	117h	—	197h	SSP2ADD	217h	—	297h	—	317h	—	397h	—
018h	LATA	098h	—	118h	—	198h	SSP2MASK	218h	—	298h	—	318h	PWM4DCL	398h	—
019h	LATB	099h	—	119h	RC1REG1	199h	SSP2STAT	219h	—	299h	—	319h	PWM4DCH	399h	—
01Ah	LATC	09Ah	—	11Ah	TX1REG1	19Ah	SSP2CON1	21Ah	—	29Ah	—	31Ah	PWM4CON	39Ah	—
01Bh	LATD(2)	09Bh	ADRESL	11Bh	SP1BRG1L	19Bh	SSP2CON2	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	LATE	09Ch	ADRESH	11Ch	SP1BRG1H	19Ch	SSP2CON3	21Ch	—	29Ch	—	31Ch	PWM5DCL	39Ch	—
01Dh	LATF(3)	09Dh	ADCON0	11Dh	RC1STA1	19Dh	—	21Dh	—	29Dh	—	31Dh	PWM5DCH	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	TX1STA1	19Eh	—	21Eh	—	29Eh	—	31Eh	PWM5CON	39Eh	—
01Fh	—	09Fh	ADACT	11Fh	BAUD1CON1	19Fh	—	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 80 Bytes	3A0h	General Purpose Register 80 Bytes
07Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
		0F0h	Common RAM Accesses 70h-7Fh	170h	Common RAM Accesses 70h-7Fh	1F0h	Common RAM Accesses 70h-7Fh	270h	Common RAM Accesses 70h-7Fh	2F0h	Common RAM Accesses 70h-7Fh	370h	Common RAM Accesses 70h-7Fh	3F0h	Common RAM Accesses 70h-7Fh
		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

Note 1: Unimplemented locations read as '0'.

2: Present only in PIC16(L)F15375/76/85/86.

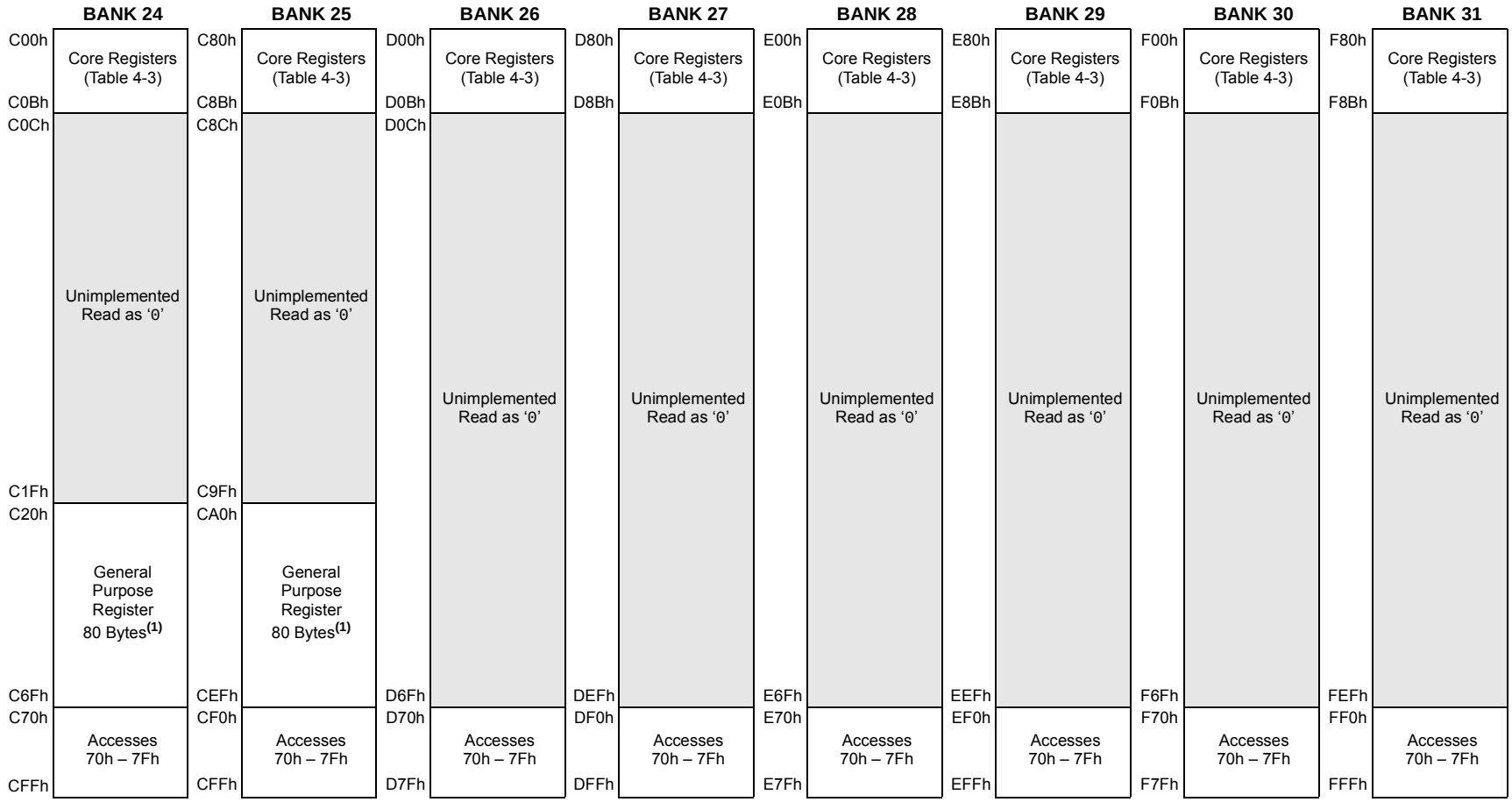
3: Present only in PIC16(L)F15385/86.

TABLE 4-6: PIC16(L)F15356/75/76/85/86 MEMORY MAP, BANKS 16-23

	BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23
800h	Core Register (Table 4-3)	880h	Core Register (Table 4-3)	900h	Core Register (Table 4-3)	980h	Core Register (Table 4-3)	A00h	Core Register (Table 4-3)	A80h	Core Register (Table 4-3)	B00h	Core Register (Table 4-3)	B80h	Core Register (Table 4-3)
80Bh		88Bh		90Bh		98Bh		A0Bh		A8Bh		B0Bh		B8Bh	
80Ch	WDTCON0	88Ch	CPUDOZE	90Ch	FVRCON	98Ch	—	A0Ch	—	A8Ch	—	B0Ch	—	B8Ch	—
80Dh	WDTCON1	88Dh	OSCCON1	90Dh	—	98Dh	—	A0Dh	—	A8Dh	—	B0Dh	—	B8Dh	—
80Eh	WDTL	88Eh	OSCCON2	90Eh	DAC1CON0	98Eh	—	A0Eh	—	A8Eh	—	B0Eh	—	B8Eh	—
80Fh	WDTH	88Fh	OSCCON3	90Fh	DAC1CON1	98Fh	CMOUT	A0Fh	—	A8Fh	—	B0Fh	—	B8Fh	—
810h	WDTU	890h	OSCSTAT1	910h	—	990h	CM1CON0	A10h	—	A90h	—	B10h	—	B90h	—
811h	BORCON	891h	OSCEN	911h	—	991h	CM1CON1	A11h	—	A91h	—	B11h	—	B91h	—
812h	VREGCON(2)	892h	OSCTUNE	912h	—	992h	CM1NCH	A12h	—	A92h	—	B12h	—	B92h	—
813h	PCON0	893h	OSCFRQ	913h	—	993h	CM1PCH	A13h	—	A93h	—	B13h	—	B93h	—
814h	PCON1	894h	—	914h	—	994h	CM2CON0	A14h	—	A94h	—	B14h	—	B94h	—
815h	—	895h	CLKRCON	915h	—	995h	CM2CON1	A15h	—	A95h	—	B15h	—	B95h	—
816h	—	896h	CLKCLK	916h	—	996h	CM2NCH	A16h	—	A96h	—	B16h	—	B96h	—
817h	—	897h	—	917h	—	997h	CM2PCH	A17h	—	A97h	—	B17h	—	B97h	—
818h	—	898h	—	918h	—	998h	—	A18h	—	A98h	—	B18h	—	B98h	—
819h	—	899h	—	919h	—	999h	—	A19h	RC2REG	A99h	—	B19h	—	B99h	—
81Ah	NVMADRL	89Ah	—	91Ah	—	99Ah	—	A1Ah	TX2REG	A9Ah	—	B1Ah	—	B9Ah	—
81Bh	NVMADRH	89Bh	—	91Bh	—	99Bh	—	A1Bh	SP2BRGL	A9Bh	—	B1Bh	—	B9Bh	—
81Ch	NVMDATL	89Ch	—	91Ch	—	99Ch	—	A1Ch	SP2BRGH	A9Ch	—	B1Ch	—	B9Ch	—
81Dh	NVMDATH	89Dh	—	91Dh	—	99Dh	—	A1Dh	RC2STA	A9Dh	—	B1Dh	—	B9Dh	—
81Eh	NVMCON1	89Eh	—	91Eh	—	99Eh	—	A1Eh	TX2STA	A9Eh	—	B1Eh	—	B9Eh	—
81Fh	NVMCON2	89Fh	—	91Fh	ZCDCON	99Fh	—	A1Fh	BAUD2CON	A9Fh	—	B1Fh	—	B9Fh	—
820h	General Purpose Register 80 Bytes(3)	8A0h	General Purpose Register 80 Bytes(3)	920h	General Purpose Register 80 Bytes(3)	9A0h	General Purpose Register 80 Bytes(3)	A20h	General Purpose Register 80 Bytes(3)	AA0h	General Purpose Register 80 Bytes(3)	B20h	General Purpose Register 80 Bytes(3)	BA0h	General Purpose Register 80 Bytes(3)
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh		BEFh	
870h	Common RAM Accesses 70h-7Fh	8F0h	Common RAM Accesses 70h-7Fh	970h	Common RAM Accesses 70h-7Fh	9F0h	Common RAM Accesses 70h-7Fh	A70h	Common RAM Accesses 70h-7Fh	AF0h	Common RAM Accesses 70h-7Fh	B70h	Common RAM Accesses 70h-7Fh	BF0h	Common RAM Accesses 70h-7Fh
87Fh		8FFh		97Fh		9FFh		A7Fh		AFFh		B7Fh		BFFh	

Note 1: Unimplemented locations read as '0'.
2: Register not implemented on LF devices.
3: Present only in PIC16(L)F15356/76/86.

TABLE 4-7: PIC16(L)F15356/75/76/85/86 MEMORY MAP, BANK 24-31



Legend: = Unimplemented data memory locations, read as '0'.

Note 1: Present only in PIC16(L)F15356/76/86.

TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (ALL BANKS)

Bank Offset Bank 0-Bank 63	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
All Banks											
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
x02h or x82h	PCL	PCL								0000 0000	0000 0000
x03h or x83h	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
x04h or x84h	FSR0L	FSR0L	Indirect Data Memory Address 0 Low Pointer							0000 0000	uuuu uuuu
x05h or x85h	FSR0H	FSR0H	Indirect Data Memory Address 0 High Pointer							0000 0000	0000 0000
x06h or x86h	FSR1L	FSR1L	Indirect Data Memory Address 1 Low Pointer							0000 0000	uuuu uuuu
x07h or x87h	FSR1H	FSR1H	Indirect Data Memory Address 1 High Pointer							0000 0000	0000 0000
x08h or x88h	BSR	—	—	BSR<5:0>						--00 0000	--00 0000
x09h or x89h	WREG	Working Register								0000 0000	uuuu uuuu
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	00-- ---1	00-- ---1

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

Note 1: These Registers can be accessed from any bank.

TABLE 4-11: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63

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 0											
	CPU CORE REGISTERS; see Table 4-10 for specifics										
00Ch	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
00Dh	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
00Eh	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
00Fh	PORTD(1)	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
010h	PORTE	—	—	—	—	RE3	RE2(1)	RE1(1)	RE0(1)	---- xxxx	---- uuuu
011h	PORTF(2)	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	xxxx xxxx	uuuu uuuu
012h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
013h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
014h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
015h	TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111
016h	TRISE	—	—	—	—	—(3)	TRISE2(1)	TRISE1(1)	TRISE0(1)	---- 1111	---- 1111
017h	TRISF(2)	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	1111 1111	1111 1111
018h	LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxx	uuuu uuuu
019h	LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	uuuu uuuu
01Ah	LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	uuuu uuuu
01Bh	LATD(1)	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	uuuu uuuu
01Ch	LATE	—	—	—	—	—	LATE2(1)	LATE1(1)	LATE0(1)	---- -xxx	---- -uuu
01Dh	LATF(2)	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx xxxx	uuuu uuuu
01Eh	—	Unimplemented								—	—
01Fh	—	Unimplemented								—	—

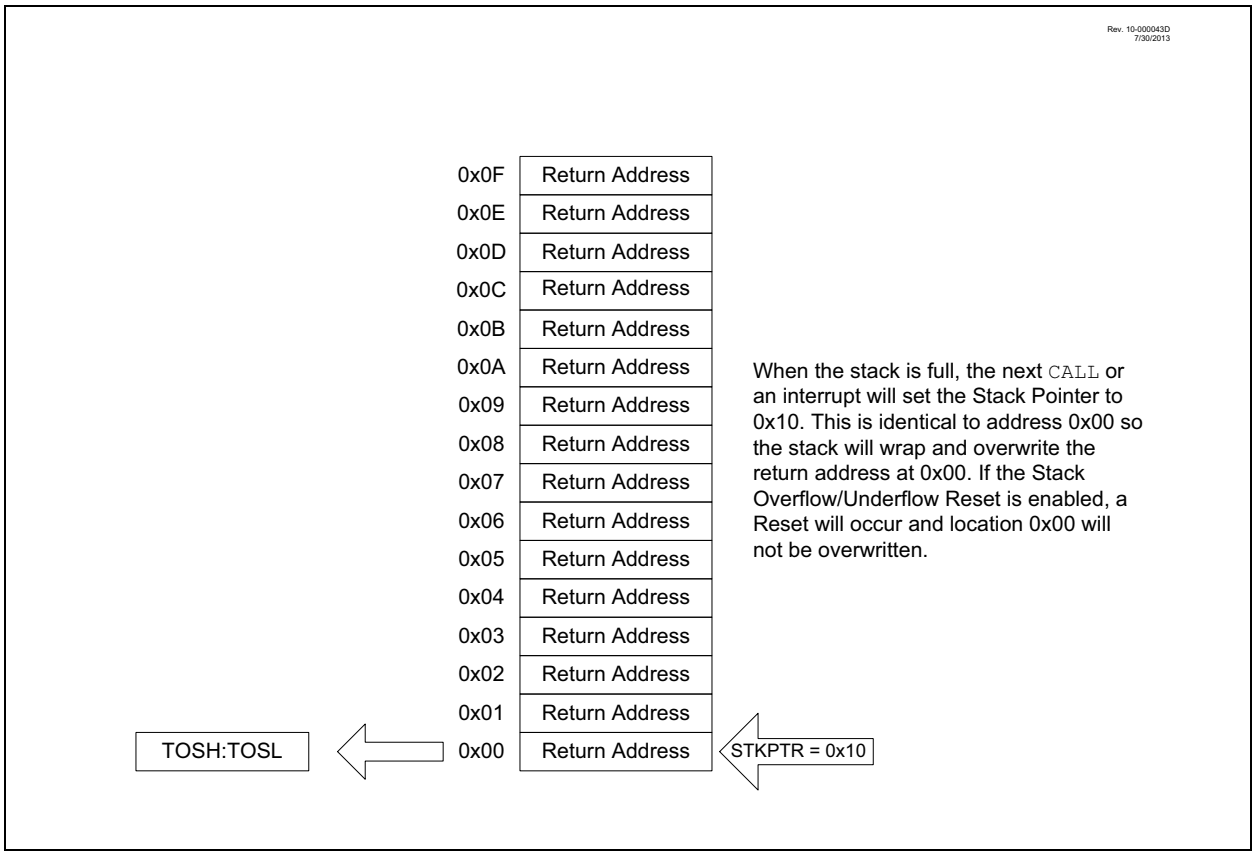
Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Present only in PIC16(L)F15375/76/85/86.

2: Present only in PIC16(L)F15385/86.

3: Unimplemented, read as '1'.

FIGURE 4-8: ACCESSING THE STACK EXAMPLE 4



4.5.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words (Register 5-2) is programmed to '1', the device will be Reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

4.6 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional/Banked Data Memory
- Linear Data Memory
- Program Flash Memory

8.13 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON registers are updated to indicate the cause of the Reset. Table 8-3 and Table 8-4 show the Reset conditions of these registers.

TABLE 8-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

STOVF	STKUNF	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{MEMV}}$	Condition
0	0	1	1	1	0	x	1	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	u	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	1	1	1	0	x	x	0	u	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
0	0	u	1	1	u	0	1	1	u	Brown-out Reset
u	u	0	u	u	u	u	0	u	u	WWDT Reset
u	u	u	u	u	u	u	0	0	u	WWDT Wake-up from Sleep
u	u	u	u	u	u	u	1	0	u	Interrupt Wake-up from Sleep
u	u	u	0	u	u	u	u	u	1	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	u	0	u	u	u	1	0	u	$\overline{\text{MCLR}}$ Reset during Sleep
u	u	u	u	0	u	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)
u	u	u	u	u	u	u	u	u	0	Memory violation Reset

TABLE 8-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON0 Register	PCON1 Register
Power-on Reset	0000h	---1 1000	0011 110x	---- --1-
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	---u uuuu	uuuu 0uuu	---- --1-
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	---1 0uuu	uuuu 0uuu	---- --u-
WWDT Timeout Reset	0000h	---0 uuuu	uuu0 uuuu	---- --u-
WWDT Wake-up from Sleep	PC + 1	---0 0uuu	uuuu uuuu	---- --u-
WWDT Window Violation	0000h	---u uuuu	uu0u uuuu	---- --u-
Brown-out Reset	0000h	---1 1000	0011 11u0	---- --u-
Interrupt Wake-up from Sleep	PC + 1(1)	---1 0uuu	uuuu uuuu	---- --u-
RESET Instruction Executed	0000h	---u uuuu	uuuu u0uu	---- --u-
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1uuu uuuu	---- --u-
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1uu uuuu	---- --u-
Memory Violation Reset ($\overline{\text{MEMV}}$ = 0)	0	-uuu uuuu	uuuu uuuu	---- --0-

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

13.4 Register Definitions: Flash Program Memory Control

REGISTER 13-1: NVMDATL: NONVOLATILE MEMORY DATA LOW BYTE REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
NVMDAT<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 **NVMDAT<7:0>**: Read/write value for Least Significant bits of program memory

REGISTER 13-2: NVMDATH: NONVOLATILE MEMORY DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	NVMDAT<13:8>					
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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **NVMDAT<13:8>**: Read/write value for Most Significant bits of program memory

REGISTER 13-3: NVMADRL: NONVOLATILE MEMORY ADDRESS 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
NVMADR<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 **NVMADR<7:0>**: Specifies the Least Significant bits for program memory address

REGISTER 13-4: NVMADRH: NONVOLATILE MEMORY ADDRESS HIGH BYTE REGISTER

U-1	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
—(1)	NVMADR<14:8>						
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 **Unimplemented:** Read as ‘1’

bit 6-0 **NVMADR<14:8>**: Specifies the Most Significant bits for program memory address

Note 1: Bit is undefined while WR = 1

REGISTER 14-5: WPUA: WEAK PULL-UP PORTA 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
WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
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 **WPUA<7:0>**: Weak Pull-up Register bits
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 14-6: ODCONA: PORTA OPEN-DRAIN CONTROL 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
ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0
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 **ODCA<7:0>:** PORTA Open-Drain Enable bits
For RA<7:0> 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)

26.0 TIMER1 MODULE WITH GATE CONTROL

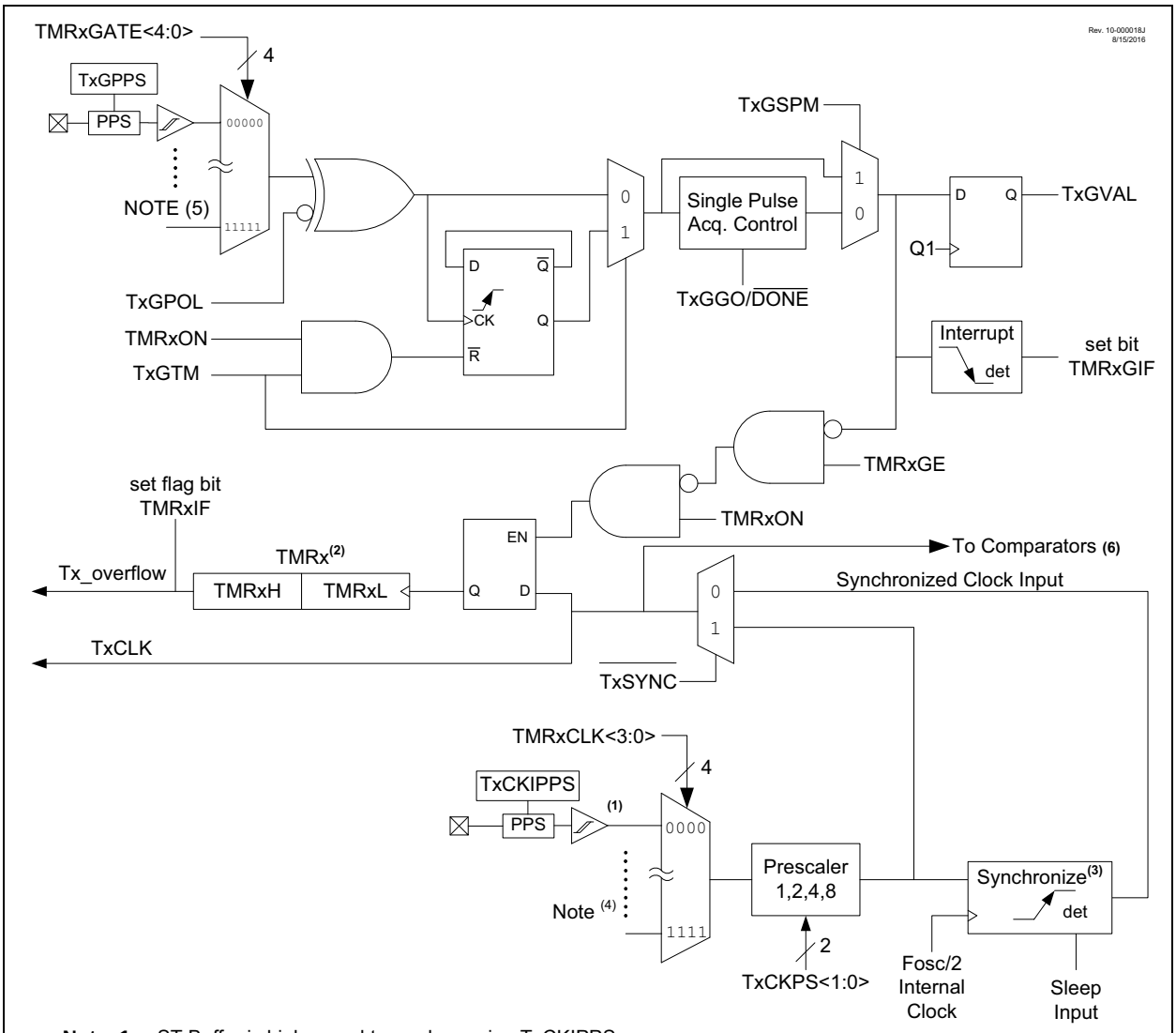
The Timer1 module is 16-bit timer/counters with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Clock source for optional comparator synchronization
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Auto-conversion Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-Pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 26-1 is a block diagram of the Timer1 module.

This device has one instance of Timer1 type modules.

FIGURE 26-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using TxCKIPPS.

2: TMRx register increments on rising edge.

3: Synchronize does not operate while in Sleep.

4: See Register 26-3 for Clock source selections.

5: See Register 26-4 for GATE source selections.

6: Synchronized comparator output should not be used in conjunction with synchronized input clock.

28.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

The Capture/Compare/PWM modules available are shown in Table 28-1.

TABLE 28-1: AVAILABLE CCP MODULES

Device	CCP1	CCP2
PIC16(L)F15356/75/76/85/86	•	•

The Capture and Compare functions are identical for all CCP modules.

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

28.1 Capture Mode

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the capture source, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCPxMODE<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIR6 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 28-1 shows a simplified diagram of the capture operation.

28.1.1 CAPTURE SOURCES

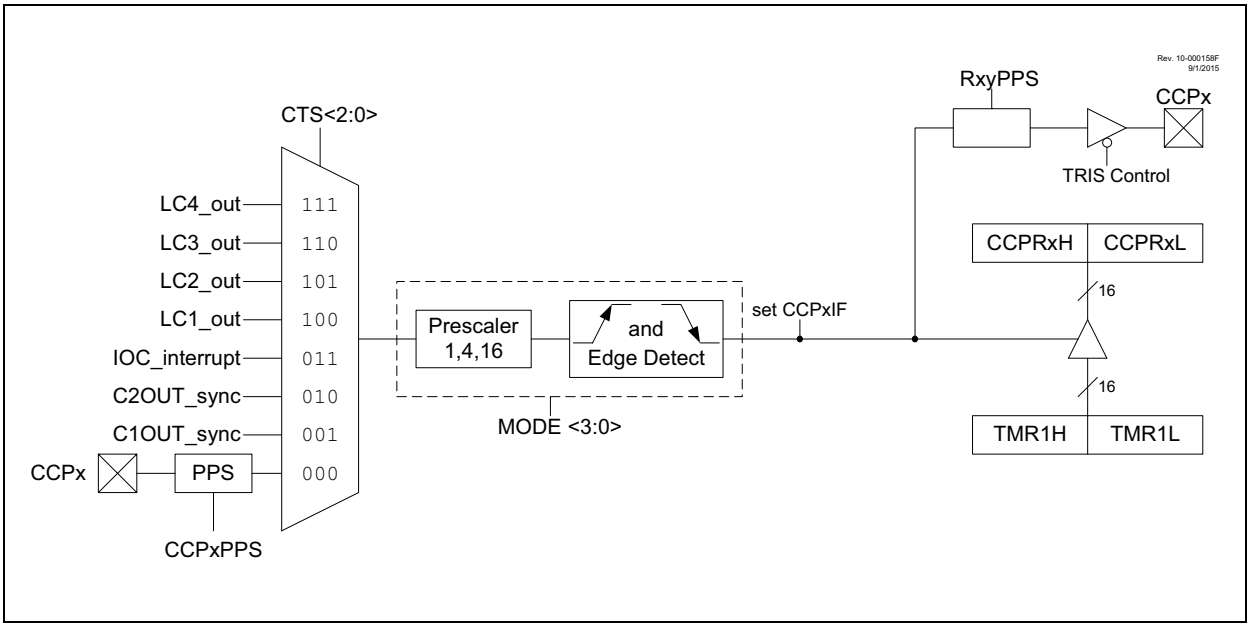
In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Note:	If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

The capture source is selected by configuring the CCPxCTS<2:0> bits of the CCPxCAP register. The following sources can be selected:

- CCPxPPS input
- C1OUT_sync
- C2OUT_sync
- IOC_interrupt
- LC1_out
- LC2_out
- LC3_out
- LC4_out

FIGURE 28-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



28.3.4 TIMER2 TIMER RESOURCE

This device has a newer version of the Timer2 module that has many new modes, which allow for greater customization and control of the PWM signals than on older parts. Refer to **Section 27.5 "Operation Examples"** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the FOSC/4 clock source selected

28.3.5 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 28-1.

EQUATION 28-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note 1: TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is transferred from the CCPRxL/H register pair into a 10-bit buffer.

Note:	The Timer postscaler (see **Section 27.4 "Timer2 Interrupt"**) is not used in the determination of the PWM frequency.

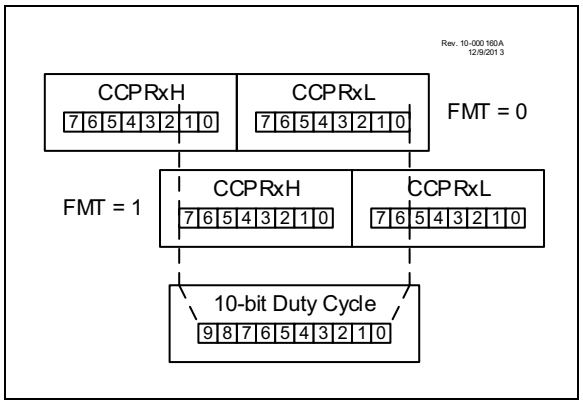
28.3.6 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the CCPRxFMT bit of the CCPxCON register (see Figure 28-5). The CCPRxH:CCPRxL register pair can be written to at any time; however the duty cycle value is not latched into the 10-bit buffer until after a match between PR2 and TMR2.

Equation 28-2 is used to calculate the PWM pulse width.

Equation 28-3 is used to calculate the PWM duty cycle ratio.

FIGURE 28-5: PWM 10-BIT ALIGNMENT



EQUATION 28-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 28-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

CCPRxH:CCPRxL register pair are used to double buffer the PWM duty cycle. This double buffering provides for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 28-4).

28.3.7 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 28-4.

EQUATION 28-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)}\ bits$$

Note:	If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

REGISTER 28-1: CCPxCON: CCPx CONTROL REGISTER (CONTINUED)

bit 3-0 **MODE<3:0>:** CCPx Mode Select bits(1)

1111 - 1100 = PWM mode (Timer2 as the timer source)
1110 = Reserved
1101 = Reserved
1100 = Reserved

1011 = Compare mode: output will pulse 0-1-0; Clears TMR1
1010 = Compare mode: output will pulse 0-1-0
1001 = Compare mode: clear output on compare match
1000 = Compare mode: set output on compare match

0111 = Capture mode: every 16th rising edge of CCPx input
0110 = Capture mode: every 4th rising edge of CCPx input
0101 = Capture mode: every rising edge of CCPx input
0100 = Capture mode: every falling edge of CCPx input

0011 = Capture mode: every edge of CCPx input
0010 = Compare mode: toggle output on match
0001 = Compare mode: toggle output on match; clear TMR1
0000 = Capture/Compare/PWM off (resets CCPx module)

Note 1: All modes will set the CCPxIF bit, and will trigger an ADC conversion if CCPx is selected as the ADC trigger source.

32.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into IDLE mode (Figure 32-30).

32.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

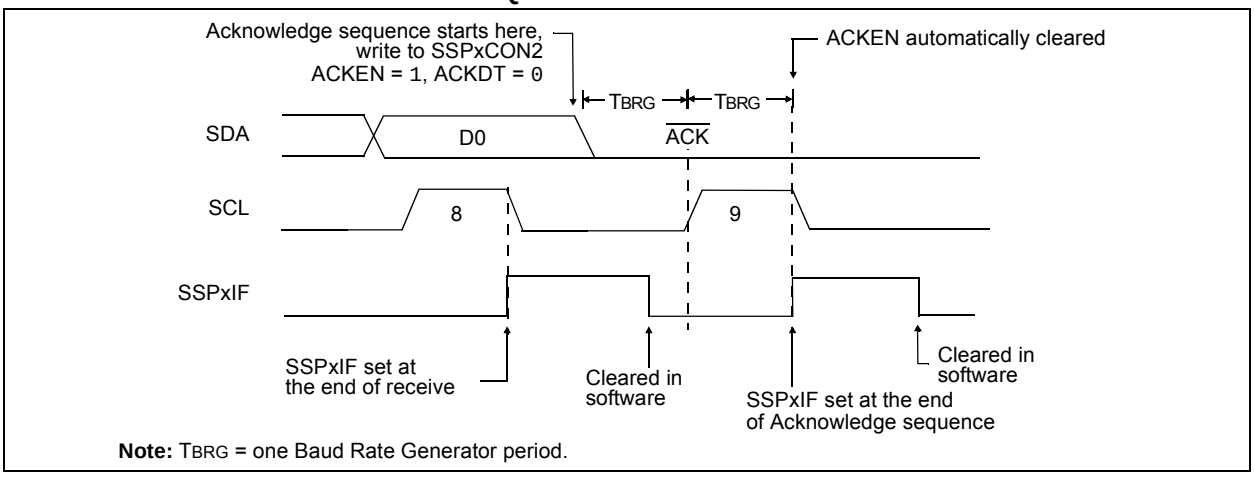
32.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 32-31).

32.6.9.1 WCOL Status Flag

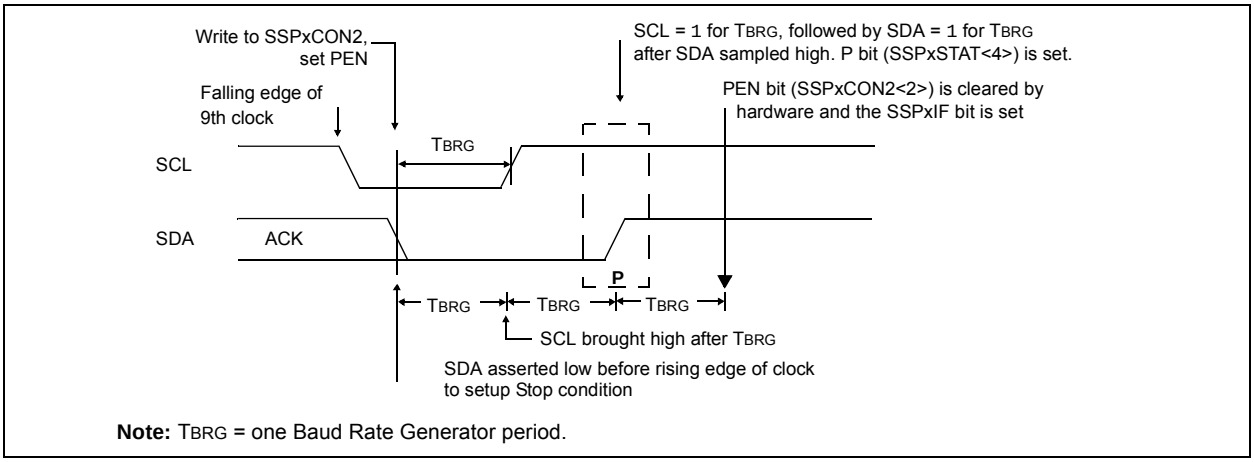
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 32-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 32-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

33.3.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDxCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPxBRGH:SPxBRGL register pair. The overflow condition will set the RXxIF flag. The counter continues to count until the fifth rising edge is detected on the RX pin. The RCIDL bit will remain false ('0') until the fifth rising edge at which time the RCIDL bit will be set. If the RCxREG is read after the overflow occurs but before the fifth rising edge then the fifth rising edge will set the RXxIF again.

Terminating the auto-baud process early to clear an overflow condition will prevent proper detection of the sync character fifth rising edge. If any falling edges of the sync character have not yet occurred when the ABDEN bit is cleared then those will be falsely detected as Start bits. The following steps are recommended to clear the overflow condition:

1. Read RCxREG to clear RXxIF.
2. If RCIDL is '0' then wait for RDCIF and repeat step 1.
3. Clear the ABDOVF bit.

33.3.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDxCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RXxIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 33-7), and asynchronously if the device is in Sleep mode (Figure 33-8). The interrupt condition is cleared by reading the RCxREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in IDLE mode waiting to receive the next character.

33.3.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be ten or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

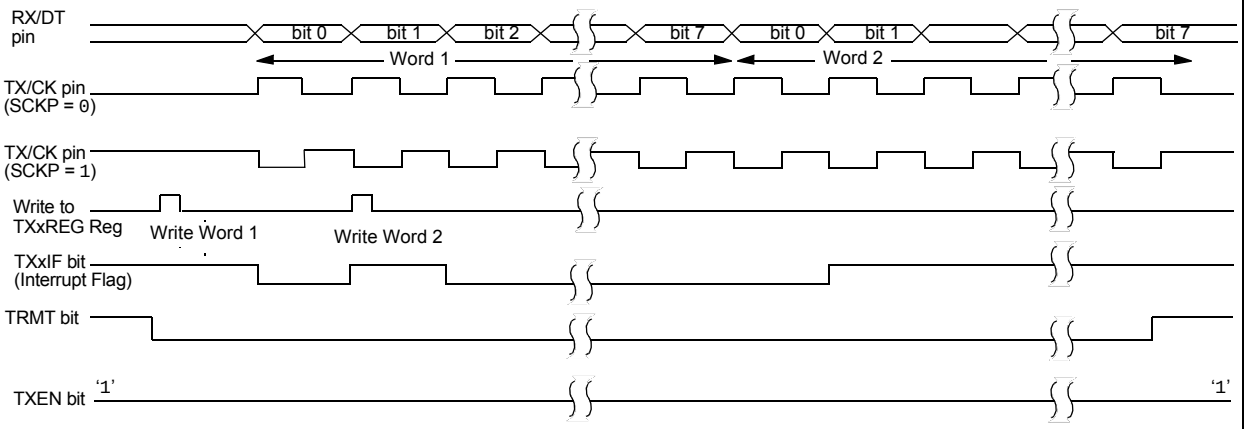
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RXxIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCxREG register and discarding its contents.

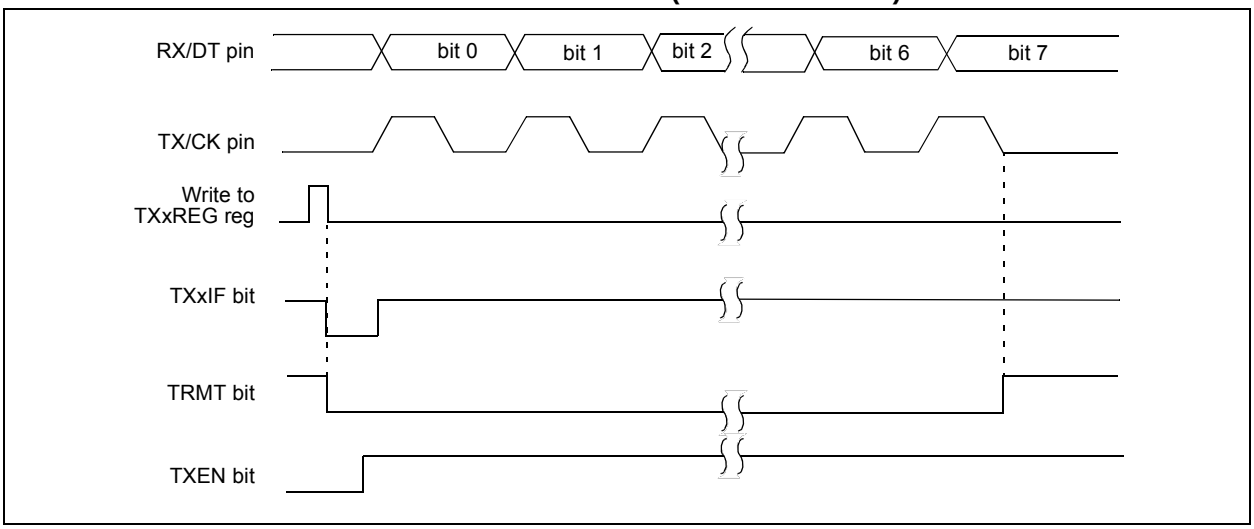
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 33-10: SYNCHRONOUS TRANSMISSION



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

FIGURE 33-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



33.4.1.5 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.

33.4.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 33.4.1.5 "Synchronous Master Reception"**), with the following exceptions:

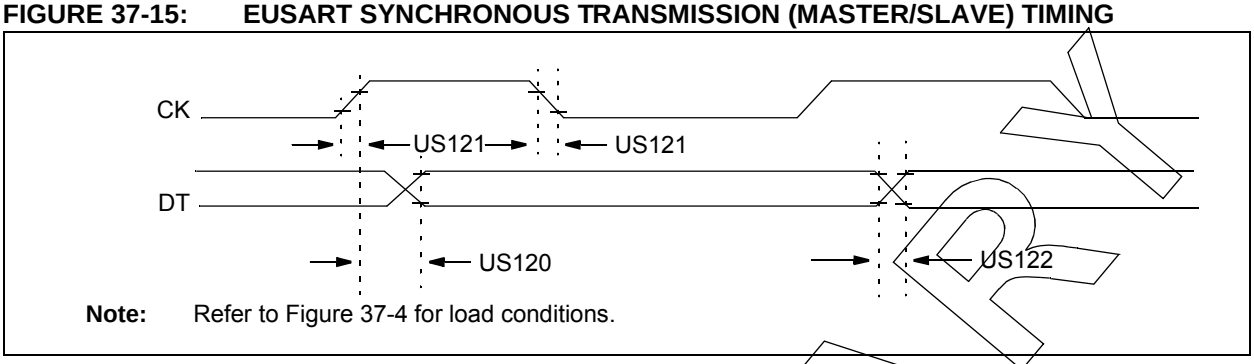
- Sleep
- CREN bit is always set, therefore the receiver is never idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCxREG register. If the RXxIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

33.4.2.4 Synchronous Slave Reception Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for both the CK and DT pins (if applicable).
3. If interrupts are desired, set the RXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RXxIF bit will be set when reception is complete. An interrupt will be generated if the RXxIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCxSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCxREG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

FIGURE 37-15: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

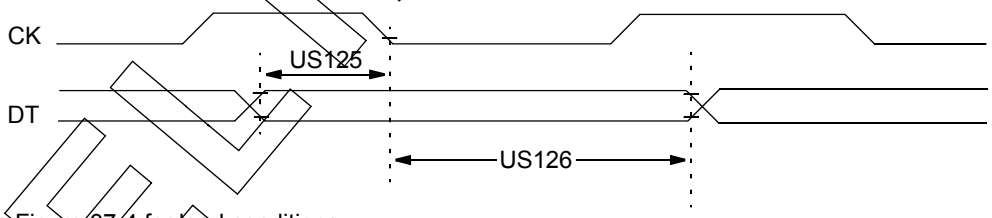


Note: Refer to Figure 37-4 for load conditions.

TABLE 37-21: EUSART SYNCHRONOUS TRANSMISSION CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US120	TCKH2DTV	SYNC XMIT (Master and Slave) Clock high to data-out valid	—	80	ns	3.0V ≤ VDD ≤ 5.5V
			—	100	ns	1.8V ≤ VDD ≤ 5.5V
US121	TCKRF	Clock out rise time and fall time (Master mode)	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V
US122	TDTRF	Data-out rise time and fall time	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V

FIGURE 37-16: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 37-4 for load conditions.

TABLE 37-22: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US125	TDTV2CKL	SYNC RCV (Master and Slave) Data-setup before CK ↓ (DT hold time)	10	—	ns	
US126	TCKL2DTL	Data-hold after CK ↓ (DT hold time)	15	—	ns	