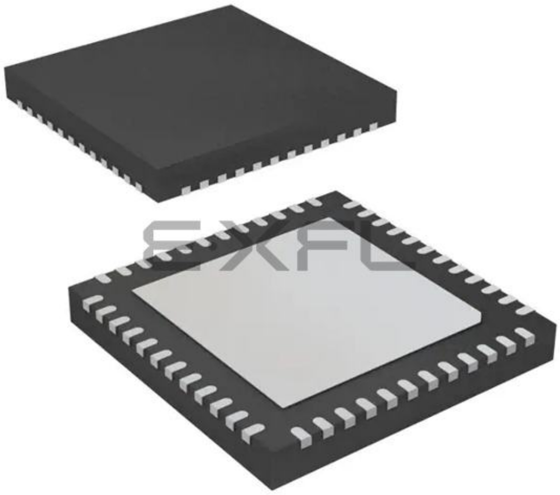


Welcome to E-XFL.COM

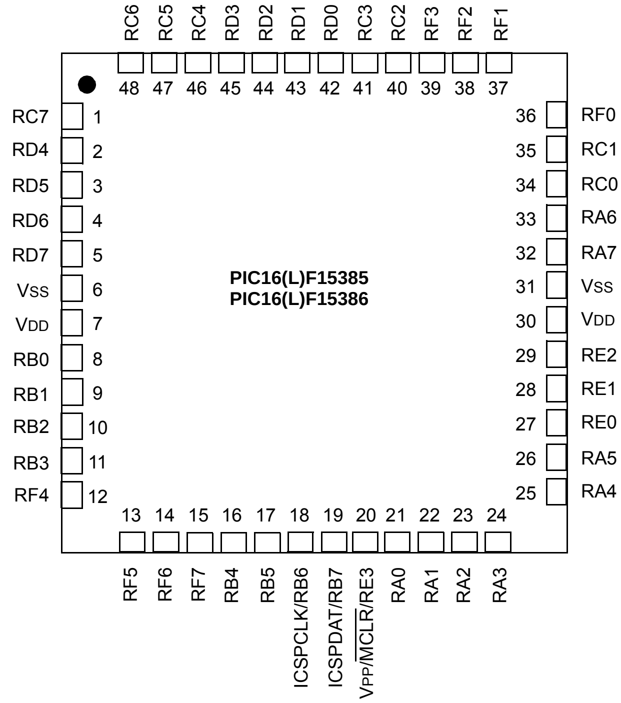
What is "Embedded - Microcontrollers"?

"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	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f15386-e-mv

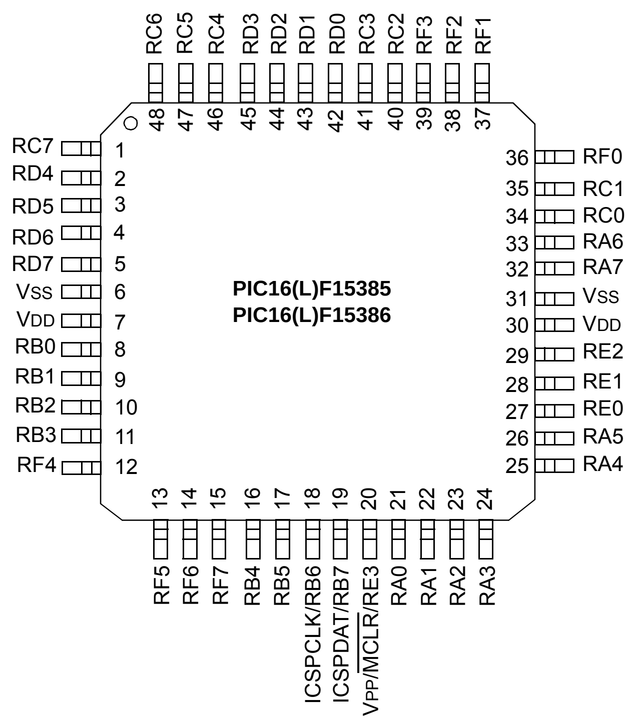
48-PIN UQFN (6x6)



Note 1: See Table 5 for location of all peripheral functions.

2: The bottom pad of the QFN/UQFN package should be connected to Vss as the circuit board level.

48-PIN TQFP (7x7)



Note: See Table 5 for location of all peripheral functions.

TABLE 1-3: PIC16(L)F15375/76 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RA6/ANA6/CLKOUT/IOCA6/OSC1	RA6	TTL/ST	CMOS/OD	General purpose I/O.
	ANA6	AN	—	ADC Channel A6 input.
	CLKOUT	—	CMOS/OD	FOSC/4 digital output (in non-crystal/resonator modes).
	IOCA6	TTL/ST	—	Interrupt-on-change input.
	OSC1	XTAL	—	External Crystal/Resonator (LP, XT, HS modes) driver input.
RA7/ANA7/CLKIN/IOCA7/OSC2	RA7	TTL/ST	CMOS/OD	General purpose I/O.
	ANA7	AN	—	ADC Channel A7 input.
	CLKIN	TTL/ST	—	External digital clock input.
	IOCA7	TTL/ST	—	Interrupt-on-change input.
	OSC2	—	XTAL	External Crystal/Resonator (LP, XT, HS modes) driver output.
RB0/ANB0/C2IN1+/ZCD1/$\overline{SS2}$(1)/CWG1(1)/INT(1)/IOCB0	RB0	TTL/ST	CMOS/OD	General purpose I/O.
	ANB0	AN	—	ADC Channel B0 input.
	C2IN1+	AN	—	Comparator positive input.
	ZCD1	AN	AN	Zero-cross detect input pin (with constant current sink/source).
	$\overline{SS2}$(1)	TTL/ST	—	MSSP2 SPI slave select input.
	CWG1(1)	TTL/ST	—	Complementary Waveform Generator 1 input.
	INT(1)	TTL/ST	—	External interrupt request input.
	IOCB0	TTL/ST	—	Interrupt-on-change input.
RB1/ANB1/C1IN3-/C2IN3-/SCL1(1)/SCK1(1)/IOCB1	RB1	TTL/ST	CMOS/OD	General purpose I/O.
	ANB1	AN	—	ADC Channel B1 input.
	C1IN3-	AN	—	Comparator negative input.
	C2IN3-	AN	—	Comparator negative input.
	SCL1(1)	I^2C	OD	MSSP1 I^2C input/output.
	SCK1(1)	TTL/ST	CMOS/OD	MSSP1 SPI clock input/output (default input location, SCK1 is a PPS remappable input and output).
	IOCB1	TTL/ST	—	Interrupt-on-change input.
RB2/ANB2/SDA1(1)/SDI1(1)/IOCB2	RB2	TTL/ST	CMOS/OD	General purpose I/O.
	ANB2	AN	—	ADC Channel B2 input.
	SDA1(1)	I^2C	OD	MSSP1 I^2C serial data input/output.
	SDI1(1)	TTL/ST	—	MSSP1 SPI serial data input (default input location, SDI1 is a PPS remappable input and output).
	IOCB2	TTL/ST	—	Interrupt-on-change input.
RB3/ANB3/C1IN2-/C2IN2-/IOCB3	RB3	TTL/ST	CMOS/OD	General purpose I/O.
	ANB3	AN	—	ADC Channel B3 input.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
	IOCB3	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 15-5, Table 15-6 and Table 15-6.

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 4-9: PIC16(L)F15356/75/76/85/86 MEMORY MAP, BANKS 60, 61, 62, AND 63

Bank 60		Bank 61		Bank 62		Bank 63	
1E3Fh	RE7PPS(2)	1EBFh	—	1F3Fh	IOCAF	1FBFh	—
1E40h	—	1EC0h	—	1F40h	—	1FC0h	—
1E41h	—	1EC1h	—	1F41h	—	1FC1h	—
1E42h	—	1EC2h	—	1F42h	—	1FC2h	—
1E43h	—	1EC3h	ADACTPPS	1F43h	ANSELB	1FC3h	—
1E44h	—	1EC4h	—	1F44h	WPUB	1FC4h	—
1E45h	—	1EC5h	SSP1CLKPPS	1F45h	ODCONB	1FC5h	—
1E46h	—	1EC6h	SSP1DATPPS	1F46h	SLRCONB	1FC6h	—
1E47h	—	1EC7h	SSP1SSPPS	1F47h	INLVLB	1FC7h	—
1E48h	—	1EC8h	SSP2CLKPPS	1F48h	IOCBP	1FC8h	—
1E49h	—	1EC9h	SSP2DATPPS	1F49h	IOCBN	1FC9h	—
1E4Ah	—	1ECAh	SSP2SSPPS	1F4Ah	IOCBF	1FCAh	—
1E4Bh	—	1ECBh	RXDT1PPS	1F4Bh	—	1FCBh	—
1E4Ch	—	1ECCh	TXCK1PPS	1F4Ch	—	1FCCh	—
1E4Dh	—	1ECDh	RXD2TPPS	1F4Dh	—	1FCDh	—
1E4Eh	—	1ECEh	TXCK2PPS	1F4Eh	ANSELC	1FCEh	—
1E4Fh	—	1ECFh	—	1F4Fh	WPUC	1FCFh	—
1E50h	ANSELF(2)	1ED0h	—	1F50h	ODCONC	1FD0h	—
1E51h	WPUF(2)	1ED1h	—	1F51h	SLRCONC	1FD1h	—
1E52h	ODCONF(2)	1ED2h	—	1F52h	INLVLC	1FD2h	—
1E53h	SLRCONF(2)	1ED3h	—	1F53h	IOCCP	1FD3h	—
1E54h	INLVLF(2)	1ED4h	—	1F54h	IOCCN	1FD4h	—
1E55h	—	1ED5h	—	1F55h	IOCCF	1FD5h	—
1E56h	—	1ED6h	—	1F56h	—	1FD6h	—
1E57h	—	1ED7h	—	1F57h	—	1FD7h	—
1E58h	—	1ED8h	—	1F58h	—	1FD8h	—
1E59h	—	1ED9h	—	1F59h	ANSELD(1)	1FD9h	—
1E5Ah	—	1EDAh	—	1F5Ah	WPUD(1)	1FDAh	—
1E5Bh	—	1EDBh	—	1F5Bh	ODCOND(1)	1FDBh	—
1E5Ch	—	1EDCh	—	1F5Ch	SLRCOND(1)	1FDCh	—
1E5Dh	—	1EDDh	—	1F5Dh	INLVLD(1)	1FDDh	—
1E5Eh	—	1EDEh	—	1F5Eh	—	1FDEh	—
1E5Fh	—	1EDFh	—	1F5Fh	—	1FDFh	—
1E60h	—	1EE0h	—	1F60h	—	1FE0h	—
1E61h	—	1EE1h	—	1F61h	—	1FE1h	—
1E62h	—	1EE2h	—	1F62h	—	1FE2h	—
1E63h	—	1EE3h	—	1F63h	—	1FE3h	BSR_ICDSHAD
1E64h	—	1EE4h	—	1F64h	ANSELE(1)	1FE4h	STATUS_SHAD
1E65h	—	1EE5h	—	1F65h	WPUE	1FE5h	WREG_SHAD
1E66h	—	1EE6h	—	1F66h	ODCONE(1)	1FE6h	BSR_SHAD
1E67h	—	1EE7h	—	1F67h	SLRCONE(1)	1FE7h	PCLATH_SHAD
1E68h	—	1EE8h	—	1F68h	INLVLE	1FE8h	FSR0L_SHAD
1E69h	—	1EE9h	—	1F69h	IOCEP	1FE9h	FSR0H_SHAD
1E6Ah	—	1EEAh	—	1F6Ah	IOCEN	1FEAh	FSR1L_SHAD
1E6Bh	—	1EEBh	—	1F6Bh	IOCEF	1FEBh	FSR1H_SHAD
1E6Ch	—	1EECh	—	1F6Ch	—	1FECh	—
1E6Dh	—	1EEDh	—	1F6Dh	—	1FEDh	STKPTR
1E6Eh	—	1EEEh	—	1F6Eh	—	1FEEh	TOSL
1E6Fh	—	1EEFh	—	1F6Fh	—	1FEFh	TOSH

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

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

2: Present only on PIC16(L)F15385/86

TABLE 4-11: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (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 14											
CPU CORE REGISTERS; see Table 4-3 for specifics											
70Ch	PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	`--00 ---0`	`--00 ---0`
70Dh	PIR1	OSFIF	CSWIF	—	—	—	—	—	ADIF	`00-- --00`	`00-- --00`
70Eh	PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	`-0-- --00`	`-0-- --00`
70Fh	PIR3	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	`0000 0000`	`0000 0000`
710h	PIR4	—	—	—	—	—	—	TMR2IF	TMR1IF	`---- --00`	`---- --00`
711h	PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	`0000 ---0`	`0000 ---0`
712h	PIR6	—	—	—	—	—	—	CCP2IF	CCP1IF	`---- --00`	`---- --00`
713h	PIR7	—	—	NVMIF	NCO1IF	—	—	—	CWG1IF	`--00 ---0`	`--00 ---0`
714h	—	Unimplemented								—	—
715h	—	Unimplemented								—	—
716h	PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	`--00 ---0`	`--00 ---0`
717h	PIE1	OSFIE	CSWIE	—	—	—	—	—	ADIE	`00-- --00`	`00-- --00`
718h	PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	`-0-- --00`	`-0-- --00`
719h	PIE3	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	`0000 0000`	`0000 0000`
71Ah	PIE4	—	—	—	—	—	—	TMR2IE	TMR1IE	`---- --00`	`---- --00`
71Bh	PIE5	CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	`0000 ---0`	`0000 ---0`
71Ch	PIE6	—	—	—	—	—	—	CCP2IE	CCP1IE	`---- --00`	`---- --00`
71Dh	PIE7	—	—	NVMIE	NCO1IE	—	—	—	CWG1IE	`--00 ---0`	`--00 ---0`
71Eh	—	Unimplemented								—	—
71Fh	—	Unimplemented								—	—

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

REGISTER 5-2: CONFIGURATION WORD 2: SUPERVISORS

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1
$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	ZCDDIS	BORV	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	U-1	U-1	U-1	R/P-1	R/P-1
BOREN1	BOREN0	$\overline{\text{LPBOREN}}$	—	—	—	$\overline{\text{PWRTE}}$	MCLRE
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	x = Bit is unknown	U = Unimplemented bit, read as ‘1’
‘0’ = Bit is cleared	‘1’ = Bit is set	W = Writable bit	n = Value when blank or after Bulk Erase

bit 13 **$\overline{\text{DEBUG}}$**: Debugger Enable bit
1 = Background debugger disabled
0 = Background debugger enabled

bit 12 **STVREN**: Stack Overflow/Underflow Reset Enable bit
1 = Stack Overflow or Underflow will cause a Reset
0 = Stack Overflow or Underflow will not cause a Reset

bit 11 **PPS1WAY**: PPSLOCK One-Way Set Enable bit
1 = The PPSLOCK bit can be cleared and set only once; PPS registers remain locked after one clear/set cycle
0 = The PPSLOCK bit can be set and cleared repeatedly (subject to the unlock sequence)

bit 10 **ZCDDIS:** Zero-Cross Detect Disable bit
1 = ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of the ZCDCON register
0 = ZCD always enabled (ZCDSEN bit is ignored)

bit 9 **BORV**: Brown-out Reset Voltage Selection bit(1)
1 = Brown-out Reset voltage (V_{BOR}) set to lower trip point level
0 = Brown-out Reset voltage (V_{BOR}) set to higher trip point level

bit 8 **Unimplemented:** Read as ‘1’

bit 7-6 **BOREN<1:0>**: Brown-out Reset Enable bits
When enabled, Brown-out Reset Voltage (V_{BOR}) is set by the BORV bit
11 = Brown-out Reset is enabled; SBOREN bit is ignored
10 = Brown-out Reset is enabled while running, disabled in Sleep; SBOREN bit is ignored
01 = Brown-out Reset is enabled according to SBOREN
00 = Brown-out Reset is disabled

bit 5 **$\overline{\text{LPBOREN}}$**: Low-Power BOR Enable bit
1 = ULPBOR is disabled
0 = ULPBOR is enabled

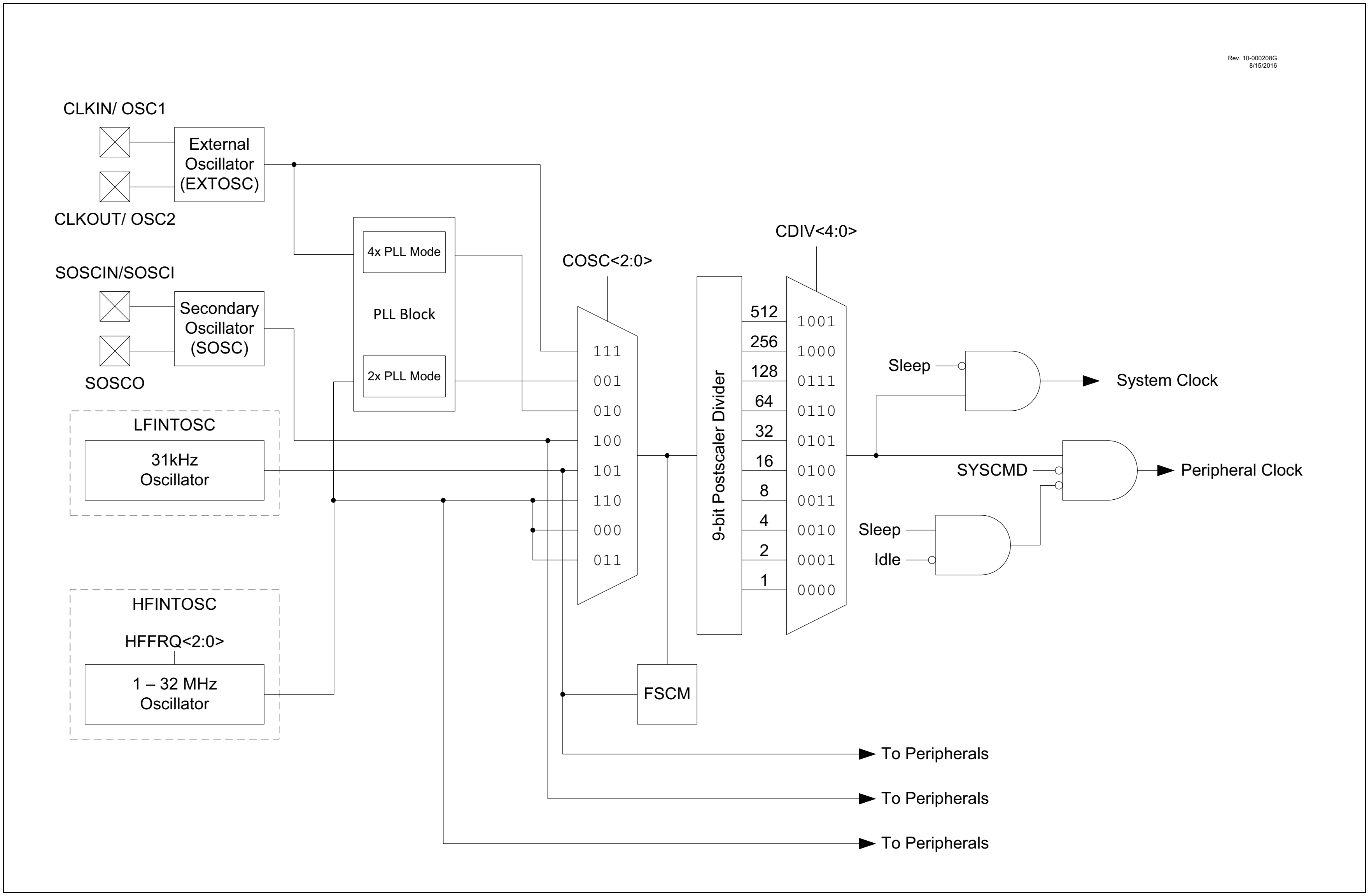
bit 4-2 **Unimplemented**: Read as ‘1’

bit 1 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit
1 = PWRT is disabled
0 = PWRT is enabled

bit 0 **MCLRE**: Master Clear ($\overline{\text{MCLR}}$) Enable bit
If LVP = 1:
RE3 pin function is $\overline{\text{MCLR}}$ (it will reset the device when driven low)
If LVP = 0:
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$ (it will reset the device when driven low)
0 = $\overline{\text{MCLR}}$ pin may be used as general purpose RE3 input

Note 1: See Vbor parameter for specific trip point voltages.
2: The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a ‘1’.

FIGURE 9-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



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

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

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

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

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

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

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

TABLE 13-2: NVM ORGANIZATION AND ACCESS INFORMATION

Master Values			NVMREG Access			FSR Access	
Memory Function	Program Counter (PC), ICSP™ Address	Memory Type	NVMREGS bit (NVMCON1)	NVMADR<14:0>	Allowed Operations	FSR Address	FSR Programming Address
Reset Vector	0000h	PFM	0	0000h	Read Write	8000h	Read-0nly
User Memory	0001h		0	0001h		8001h	
	0003h			0003h		8003h	
INT Vector	0004h		0	0004h		8004h	
User Memory	0005h		0	0005h		8005h	
	1FFFh			1FFFh		9FFFh	
	3FFFh			3FFFh		BFFFh	
User ID	8000h	PFM	1	0000h	Read	No Access	
	8003h			0003h	Write		
Reserved	8004h	—	–	0004h	—		
Rev ID	8005h	PFM	1	0005h	Read-Only		
Device ID	8006h		1	0006h			
CONFIG1	8007h		1	0007h	Read Write		
CONFIG2	8008h		1	0008h			
CONFIG3	8009h		1	0009h			
CONFIG4	800Ah		1	000Ah			
CONFIG5	800Bh		1	000Bh			
DIA and DCI	8100h-82FFh	PFM and Hard coded	1	0100h-02FFh	Read-Only	No Access	

TABLE 15-6: PPS OUTPUT SIGNAL ROUTING OPTIONS (PIC16(L)F15375/76)

Output Signal Name	RxyPPS Register Value	Remappable to Pins of PORTx				
		PIC16(L)F15375/76				
		PORTA	PORTB	PORTC	PORTD	PORTE
CLKR	0x1B		•	•		
NCO1OUT	0x1A	•			•	
TMR0	0x19		•	•		
SDO2/SDA2	0x18		•		•	
SCK2/SCL2	0x17		•		•	
SDO1/SDA1	0x16		•	•		
SCK1/SCL1	0x15		•	•		
C2OUT	0x14	•				•
C1OUT	0x13	•			•	
DT2	0x12		•		•	
TX2/CK2	0x11		•		•	
DT1	0x10		•	•		
TX1/CK1	0x0F		•	•		
PWM6OUT	0x0E	•			•	
PWM5OUT	0x0D	•		•		
PWM4OUT	0x0C		•		•	
PWM3OUT	0x0B		•		•	
CCP2	0x0A		•	•		
CCP1	0x09		•	•		
CWG1D	0x08		•		•	
CWG1C	0x07		•		•	
CWG1B	0x06		•		•	
CWG1A	0x05		•	•		
CLC4OUT	0x04		•		•	
CLC3OUT	0x03		•		•	
CLC2OUT	0x02	•		•		
CLC1OUT	0x01	•		•		

20.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 20-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

FIGURE 20-1: ADC BLOCK DIAGRAM

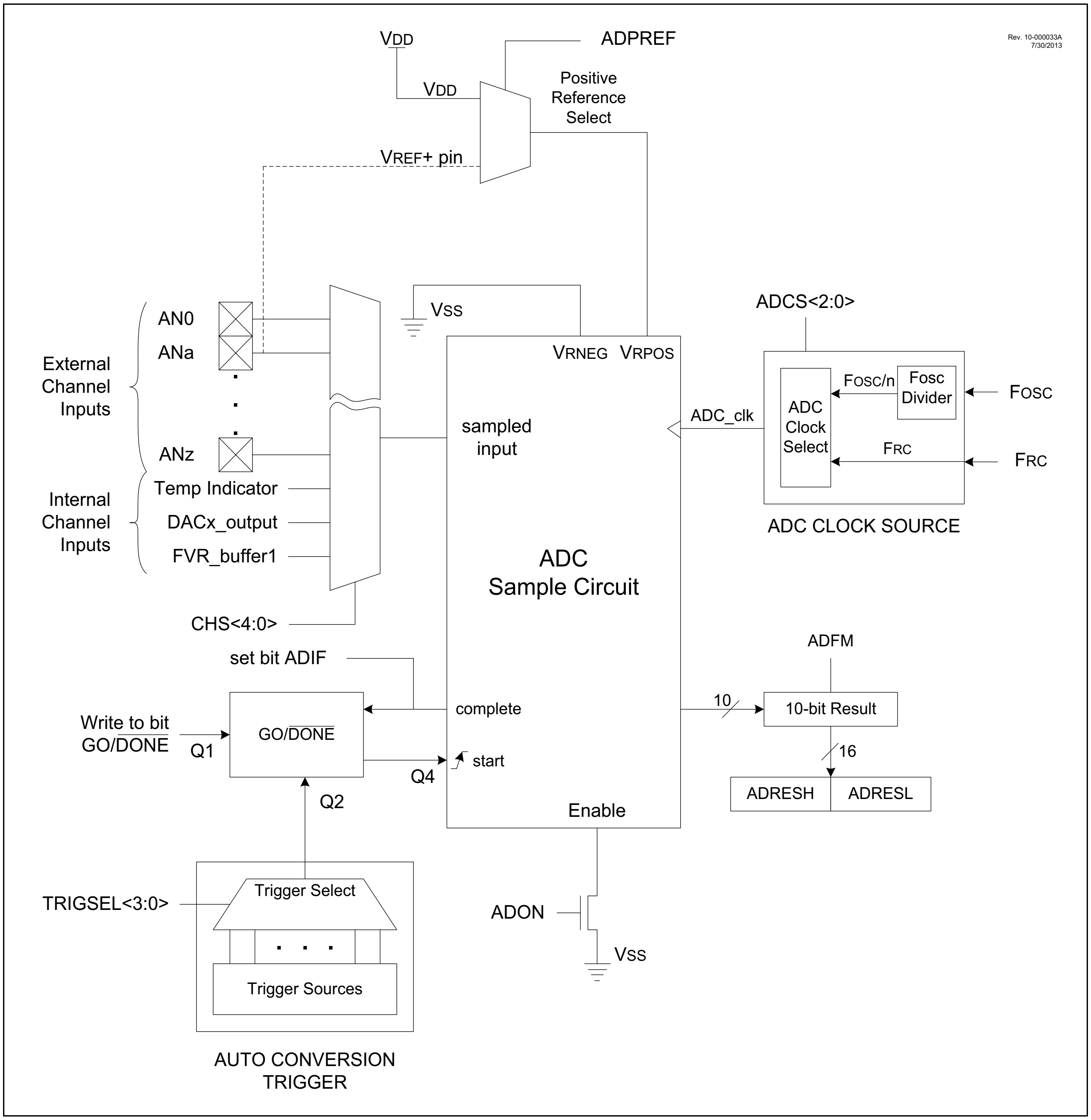
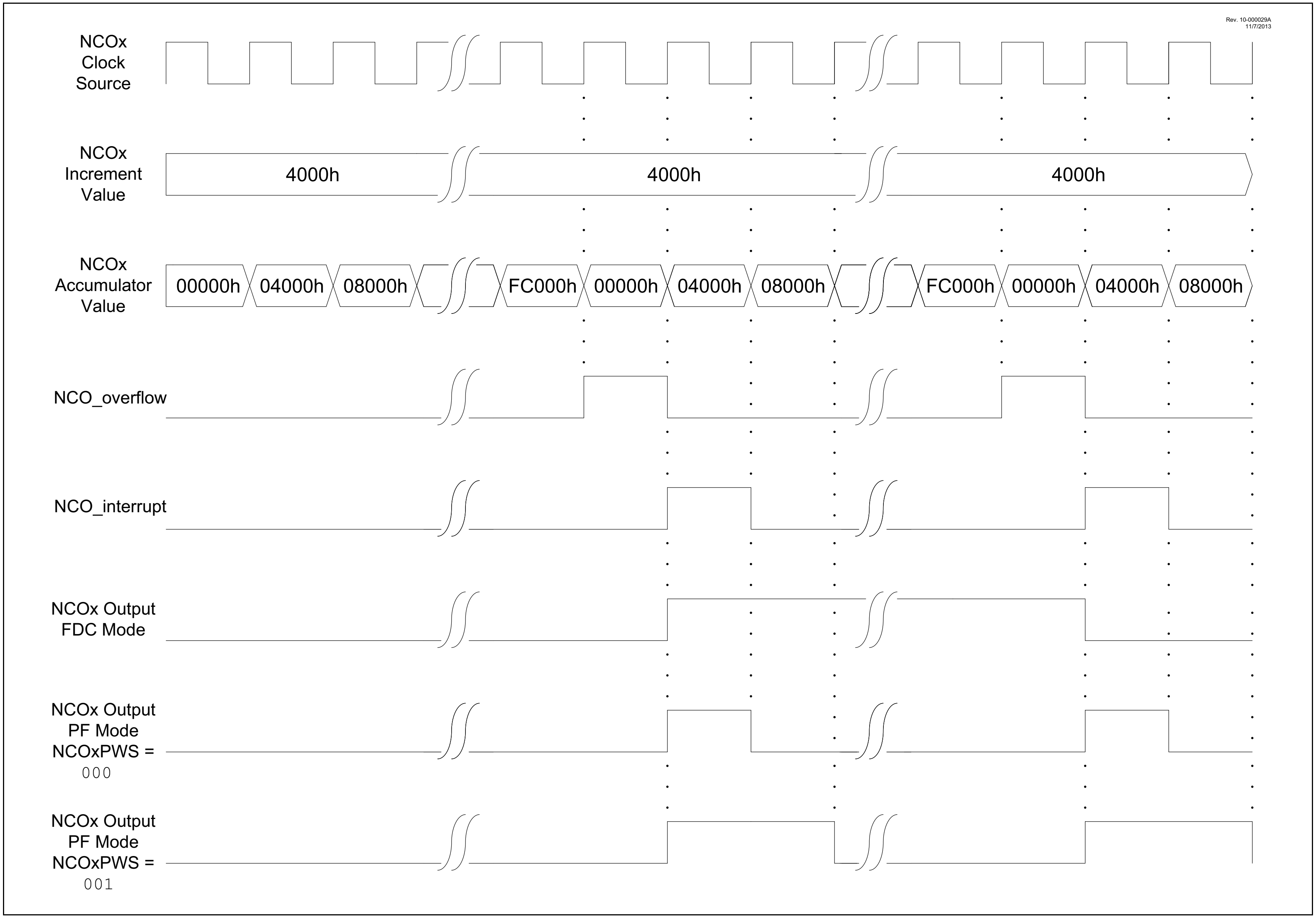


FIGURE 22-2: FDC OUTPUT MODE OPERATION DIAGRAM



24.9 Register Definitions: ZCD Control

REGISTER 24-1: ZCDCON: ZERO-CROSS DETECTION CONTROL REGISTER

R/W-q/q	U-0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
SEN	—	OUT	POL	—	—	INTP	INTN
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	q = value depends on Configuration bits

bit 7 **SEN**: Zero-Cross Detection Enable bit
1 = Zero-cross detect is enabled. ZCD pin is forced to output to source and sink current.
0 = Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **OUT:** Zero-Cross Detection Logic Level bit
POL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current
POL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current

bit 4 **POL:** Zero-Cross Detection Logic Output Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **INTP:** Zero-Cross Positive Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCDx_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCDx_output transition

bit 0 **INTN:** Zero-Cross Negative Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCDx_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCDx_output transition

TABLE 24-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	150
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	158
ZCDxCON	EN	—	OUT	POL	—	—	INTP	INTN	314

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

TABLE 24-2: SUMMARY OF CONFIGURATION WORD WITH THE ZCD MODULE

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG2	13:8	—	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	ZCDDIS	BORV	—	103
	7:0	BOREN <1:0>		$\overline{\text{LPBOREN}}$	—	—	—	$\overline{\text{PWRTE}}$	MCLRE	

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

REGISTER 26-4: T1GATE TIMER1 GATE SELECT REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	GSS<4:0>				
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	HC = Bit is cleared by hardware

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **GSS<4:0>:** Timer1 Gate Select bits

11111-10001 = Reserved
10000 = LC4_out
01111 = LC3_out
01110 = LC2_out
01101 = LC1_out
00100 = ZCD1_output
01011 = C2OUT_sync
01010 = C1OUT_sync
01001 = NCO1_out
01000 = PWM6_out
00111 = PWM5_out
00110 = PWM4_out
00101 = PWM3_out
00100 = CCP2_out
00011 = CCP1_out
00010 = TMR2_postscaled
00001 = Timer0 overflow output
00000 = T1GPPS

27.5.7 EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE

In Edge-Triggered Hardware Limit One-Shot modes the timer starts on the first external signal edge after the ON bit is set and resets on all subsequent edges. Only the first edge after the ON bit is set is needed to start the timer. The counter will resume counting automatically two clocks after all subsequent external Reset edges. Edge triggers are as follows:

- Rising edge start and Reset (MODE<4:0> = `01100`)
- Falling edge start and Reset (MODE<4:0> = `01101`)

The timer resets and clears the ON bit when the timer value matches the PRx period value. External signal edges will have no effect until after software sets the ON bit. Figure 27-10 illustrates the rising edge hardware limit one-shot operation.

When this mode is used in conjunction with the CCP then the first starting edge trigger, and all subsequent Reset edges, will activate the PWM drive. The PWM drive will deactivate when the timer matches the CCPRx pulse-width value and stay deactivated until the timer halts at the PRx period match unless an external signal edge resets the timer before the match occurs.

REGISTER 27-3: T2HLT: TIMERx HARDWARE LIMIT 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
PSYNC(1, 2)	CKPOL(3)	CKSYNC(4, 5)	MODE<4:0>(6, 7)				
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 **PSYNC:** Timerx Prescaler Synchronization Enable bit(1, 2)
1 = TMRx Prescaler Output is synchronized to Fosc/4
0 = TMRx Prescaler Output is not synchronized to Fosc/4

bit 6 **CKPOL:** Timerx Clock Polarity Selection bit(3)
1 = Falling edge of input clock clocks timer/prescaler
0 = Rising edge of input clock clocks timer/prescaler

bit 5 **CKSYNC:** Timerx Clock Synchronization Enable bit(4, 5)
1 = ON register bit is synchronized to TMR2_clk input
0 = ON register bit is not synchronized to TMR2_clk input

bit 4-0 **MODE<4:0>:** Timerx Control Mode Selection bits(6, 7)
See Table 27-1.

Note 1: Setting this bit ensures that reading TMRx will return a valid value.
2: When this bit is ‘1’, Timer2 cannot operate in Sleep mode.
3: CKPOL should not be changed while ON = 1.
4: Setting this bit ensures glitch-free operation when the ON is enabled or disabled.
5: When this bit is set then the timer operation will be delayed by two TMRx input clocks after the ON bit is set.
6: Unless otherwise indicated, all modes start upon ON = 1 and stop upon ON = 0 (stops occur without affecting the value of TMRx).
7: When TMRx = PRx, the next clock clears TMRx, regardless of the operating mode.

REGISTER 31-11: CLCDATA: CLC DATA OUTPUT

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT
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 **MLC4OUT:** Mirror copy of LC4OUT bit

bit 2 **MLC3OUT:** Mirror copy of LC3OUT bit

bit 1 **MLC2OUT:** Mirror copy of LC2OUT bit

bit 0 **MLC1OUT:** Mirror copy of LC1OUT bit

32.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level (Case 1).
b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 32-36). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 32-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 32-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

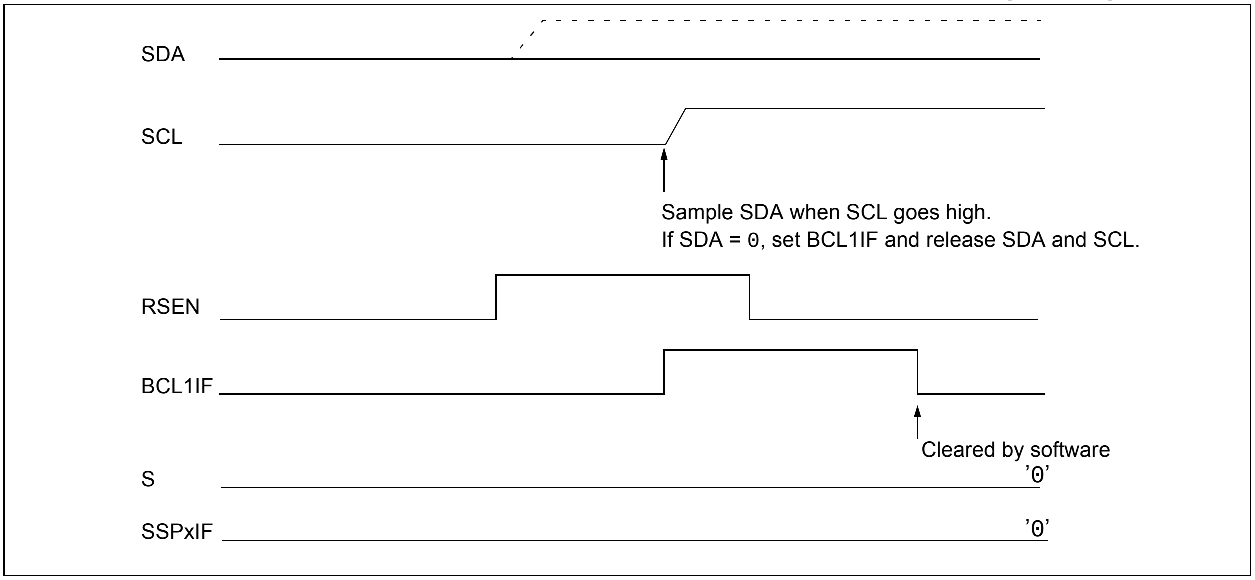
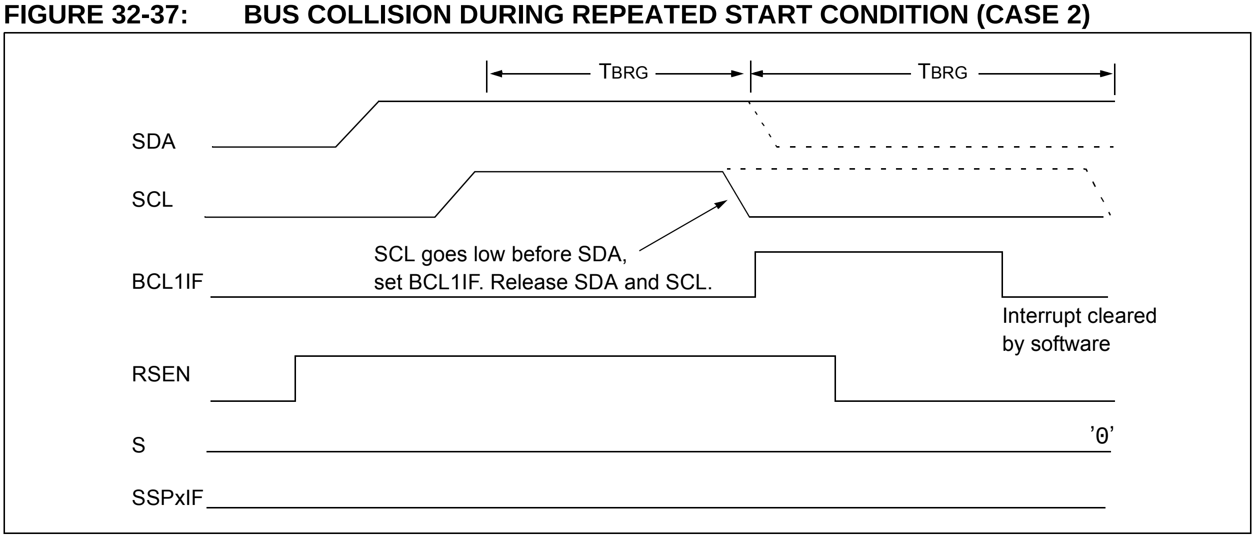


FIGURE 32-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



33.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V_{OH} Mark state which represents a '1' data bit, and a V_{OL} Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 33-3 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

33.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 33-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXxREG register.

33.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXxSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXxSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCxSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note:	The TXxIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

33.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXxREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXxREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXxREG until the Stop bit of the previous character has been transmitted. The pending character in the TXxREG is then transferred to the TSR in one T_{CY} immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXxREG.

33.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDxCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 33.4.1.2 "Clock Polarity"**.

33.1.1.4 Transmit Interrupt Flag

The TXxIF interrupt flag bit of the PIR3 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXxREG. In other words, the TXxIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXxREG. The TXxIF flag bit is not cleared immediately upon writing TXxREG. TXxIF becomes valid in the second instruction cycle following the write execution. Polling TXxIF immediately following the TXxREG write will return invalid results. The TXxIF bit is read-only, it cannot be set or cleared by software.

The TXxIF interrupt can be enabled by setting the TXxIE interrupt enable bit of the PIE3 register. However, the TXxIF flag bit will be set whenever the TXxREG is empty, regardless of the state of TXxIE enable bit.

To use interrupts when transmitting data, set the TXxIE bit only when there is more data to send. Clear the TXxIE interrupt enable bit upon writing the last character of the transmission to the TXxREG.

33.1.2.8 Asynchronous Reception Setup:

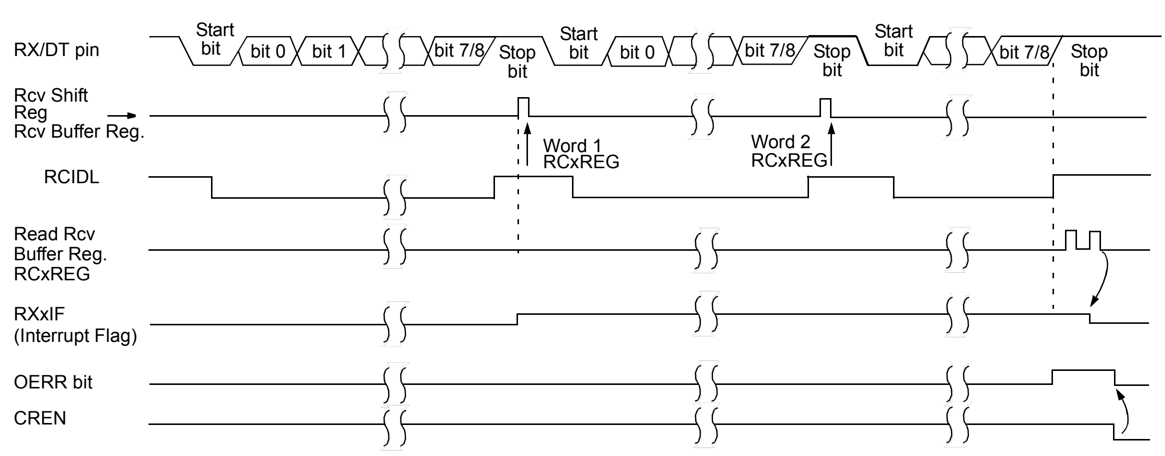
1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 33.3 "EUSART Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set the RX9 bit.
6. Enable reception by setting the CREN bit.
7. The RXxIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RXxIE interrupt enable bit was also set.
8. Read the RCxSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
9. Get the received eight Least Significant data bits from the receive buffer by reading the RCxREG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

33.1.2.9 9-bit Address Detection Mode Setup

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 33.3 "EUSART Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
5. Enable 9-bit reception by setting the RX9 bit.
6. Enable address detection by setting the ADDEN bit.
7. Enable reception by setting the CREN bit.
8. The RXxIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RXxIE interrupt enable bit was also set.
9. Read the RCxSTA register to get the error flags. The ninth data bit will always be set.
10. Get the received eight Least Significant data bits from the receive buffer by reading the RCxREG register. Software determines if this is the device's address.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 33-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCxREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

MOVWI	Move W to INDFn
Syntax:	[*label*] MOVWI ++FSRn [*label*] MOVWI --FSRn [*label*] MOVWI FSRn++ [*label*] MOVWI FSRn-- [*label*] MOVWI k[FSRn]
Operands:	n ∈ [0,1] mm ∈ [00,01, 10, 11] $-32 \leq k \leq 31$
Operation:	W → INDFn Effective address is determined by • FSR + 1 (preincrement) • FSR - 1 (predecrement) • FSR + k (relative offset) After the Move, the FSR value will be either: • FSR + 1 (all increments) • FSR - 1 (all decrements) Unchanged
Status Affected:	None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h-FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

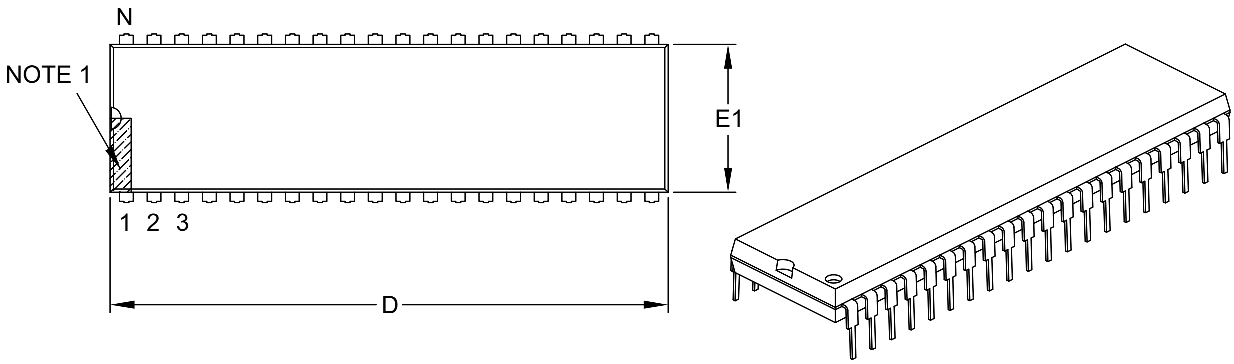
NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	`NOP`

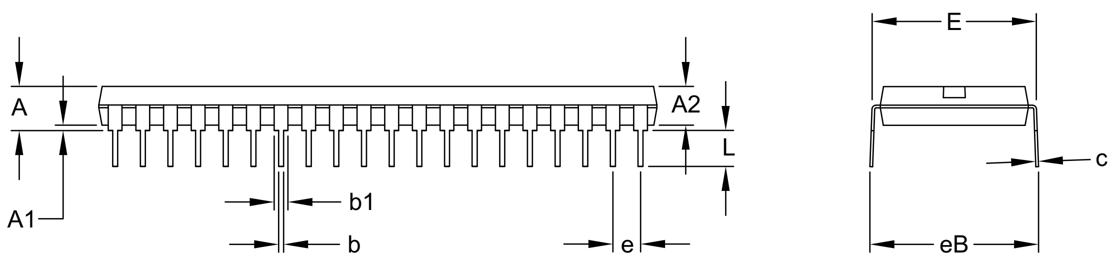
RESET	Software Reset
Syntax:	[*label*] RESET
Operands:	None
Operation:	Execute a device Reset. Resets the $\overline{\text{RI}}$ flag of the PCON register.
Status Affected:	None
Description:	This instruction provides a way to execute a hardware Reset by software.

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE k
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words:	1
Cycles:	2
Example:	`RETFIE` After Interrupt PC = TOS GIE = 1

40-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging





Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		40	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.250
Molded Package Thickness	A2	.125	–	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.590	–	.625
Molded Package Width	E1	.485	–	.580
Overall Length	D	1.980	–	2.095
Tip to Seating Plane	L	.115	–	.200
Lead Thickness	c	.008	–	.015
Upper Lead Width	b1	.030	–	.070
Lower Lead Width	b	.014	–	.023
Overall Row Spacing §	eB	–	–	.700

Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-016B

