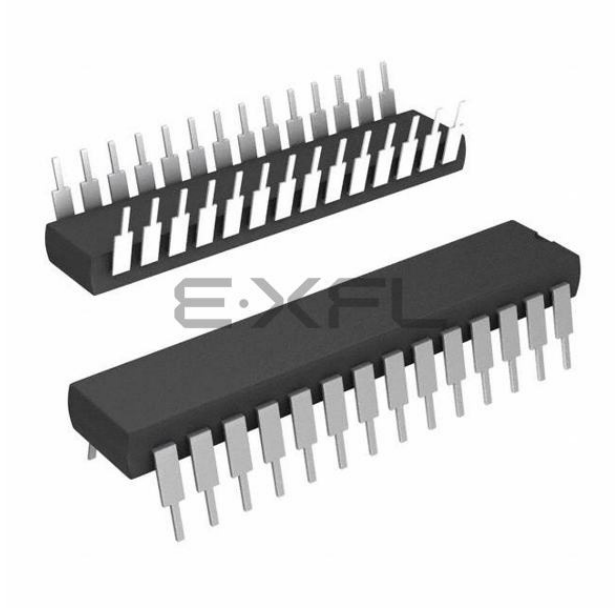


Welcome to E-XFL.COM

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	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f25j11-i-sp

TABLE 1-3: PIC18F2XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	28-SPDIP/ SSOP/ SOIC	28-QFN			
					PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN12/INT0/RP3	21	18			
RB0			I/O	DIG	Digital I/O.
AN12			I	Analog	Analog input 12.
INT0			I	ST	External interrupt 0.
RP3			I/O	DIG	Remappable peripheral pin 3.
RB1/AN10/RTCC/RP4	22	19			
RB1			I/O	DIG	Digital I/O.
AN10			I	Analog	Analog input 10.
RTCC			O	DIG	Real Time Clock Calendar output.
RP4			I/O	DIG	Remappable peripheral pin 4.
RB2/AN8/CTED1/ REFO/RP5	23	20			
RB2			I/O	DIG	Digital I/O.
AN8			I	Analog	Analog input 8.
CTED1			I	ST	CTMU edge 1 input.
REFO			O	DIG	Reference output clock.
RP5			I/O	DIG	Remappable peripheral pin 5.
RB3/AN9/CTED2/RP6	24	21			
RB3			I/O	DIG	Digital I/O.
AN9			I	Analog	Analog input 9.
CTED2			I/O	ST	CTMU edge 2 input.
RP6			I	DIG	Remappable peripheral pin 6.
RB4/KBI0/RP7	25	22			
RB4			I/O	DIG	Digital I/O.
KBI0			I	TTL	Interrupt-on-change pin.
RP7			I/O	DIG	Remappable peripheral pin 7.
RB5/KBI1/RP8	26	23			
RB5			I/O	DIG	Digital I/O.
KBI1			I	TTL	Interrupt-on-change pin.
RP8			I/O	DIG	Remappable peripheral pin 8.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD}); DIG = Digital output

Note 1: RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

© 2011 Microchip Technology Inc.

3.5 Effects of Power-Managed Modes on Various Clock Sources

When the PRI_IDLE mode is selected, the designated primary oscillator continues to run without interruption. In secondary clock modes (SEC_RUN and SEC_IDLE), the Timer1 oscillator is operating and providing the device clock. The Timer1 oscillator may also run in all power-managed modes if required to clock Timer1 or Timer3.

In internal oscillator modes (RC_RUN and RC_IDLE), the internal oscillator block provides the device clock source. The 31 kHz INTRC output can be used directly to provide the clock and may be enabled to support various special features regardless of the power-managed mode (see **Section 26.2 "Watchdog Timer (WDT)"**, **Section 26.4 "Two-Speed Start-up"** and **Section 26.5 "Fail-Safe Clock Monitor"** for more information on WDT, FSCM and Two-Speed Start-up). The INTOSC output at 8 MHz may be used directly to clock the device or may be divided down by the postscaler. The INTOSC output is disabled if the clock is provided directly from the INTRC output.

If Sleep mode is selected, all clock sources, which are no longer required, are stopped. Since all the transistor switching currents have been stopped, Sleep mode achieves the lowest current consumption of the device (only leakage currents) outside of Deep Sleep mode.

Enabling any on-chip feature that will operate during Sleep mode increases the current consumed during Sleep mode. The INTRC is required to support WDT operation. The Timer1 oscillator may be operating to support an RTC. Other features may be operating that do not require a device clock source (i.e., MSSP slave, PMP, INTx pins, etc.). Peripherals that may add significant current consumption are listed in **Section 29.2 "DC Characteristics: Power-Down and Supply Current PIC18F46J11 Family (Industrial)"**.

3.6 Power-up Delays

Power-up delays are controlled by two timers so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 5.6 "Power-up Timer (PWRT)"**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up (parameter 33, Table 29-15).

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (HS mode). The OST does this by counting 1024 oscillator cycles before allowing the oscillator to clock the device.

There is a delay of interval, T_{CSD} (parameter 38, Table 29-15), following POR, while the controller becomes ready to execute instructions. This delay runs concurrently with any other delays. This may be the only delay that occurs when any of the internal oscillator or EC modes are used as the primary clock source.

© 2011 Microchip Technology Inc.

6.2.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as 2 bytes or 4 bytes in program memory. The Least Significant Byte (LSB) of an instruction word is always stored in a program memory location with an even address (LSB = 0). To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read '0' (see **Section 6.1.3 "Program Counter"**).

Figure 6-5 provides an example of how instruction words are stored in the program memory.

The CALL and GOTO instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 6-5 displays how the instruction, GOTO 0006h, is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 27.0 "Instruction Set Summary"** provides further details of the instruction set.

FIGURE 6-5: INSTRUCTIONS IN PROGRAM MEMORY

Program Memory Byte Locations →

			LSB = 1	LSB = 0	Word Address ↓
					000000h
					000002h
					000004h
					000006h
Instruction 1:	MOVLW	055h	0Fh	55h	000008h
Instruction 2:	GOTO	0006h	EFh	03h	00000Ah
			F0h	00h	00000Ch
Instruction 3:	MOVFF	123h, 456h	C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h

6.2.4 TWO-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: CALL, MOVFF, GOTO and LSFR. In all cases, the second word of the instructions always has '1111' as its four Most Significant bits (MSbs); the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the 4 MSbs of an instruction specifies a special form of NOP. If the instruction is executed in proper sequence immediately after the first word, the data in the second word is accessed and used by the instruction sequence. If the first word is skipped for some reason, and the second word is executed by itself, a NOP is executed instead. This is necessary for cases when the two-word instruction is preceded by a conditional instruction that changes the PC. Example 6-4 illustrates how this works.

Note:	See **Section 6.5 "Program Memory and the Extended Instruction Set"** for information on two-word instructions in the extended instruction set.

EXAMPLE 6-4: TWO-WORD INSTRUCTIONS

CASE 1:

Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; No, skip this word
1111 0100 0101 0110			; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3	; continue code

CASE 2:

Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; Yes, execute this word
1111 0100 0101 0110			; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3	; continue code

7.5.3 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

7.5.4 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a MCLR Reset or a WDT time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

7.6 Flash Program Operation During Code Protection

See **Section 26.6 "Program Verification and Code Protection"** for details on code protection of Flash program memory.

TABLE 7-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					69
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								69
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								69
TABLAT	Program Memory Table Latch								69
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
EECON2	Program Memory Control Register 2 (not a physical register)								71
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	71

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash program memory access.

© 2011 Microchip Technology Inc.

9.1 INTCON Registers

The INTCON registers are readable and writable registers, which contain various enable, priority and flag bits.

Note:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER (ACCESS FF2h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1 and GIEH = 1:
1 = Enables all low-priority peripheral interrupts
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB<7:4> pins changed state (must be cleared in software)
0 = None of the RB<7:4> pins have changed state

Note 1: A mismatch condition will continue to set this bit. Reading PORTB and waiting 1 TCY will end the mismatch condition and allow the bit to be cleared.

10.7.6 PERIPHERAL PIN SELECT REGISTERS

The PIC18F46J11 family of devices implements a total of 37 registers for remappable peripheral configuration of 44-pin devices. The 28-pin devices have 31 registers for remappable peripheral configuration.

Note: Input and output register values can only be changed if PPS<IOLOCK> = 0. See Example 10-7 for a specific command sequence.

REGISTER 10-5: PPSCON: PERIPHERAL PIN SELECT INPUT REGISTER 0 (BANKED EFFh)(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IOLOCK
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **IOLOCK:** I/O Lock Enable bit

1 = I/O lock active, RPORx and RPINRx registers are write-protected
0 = I/O lock not active, pin configurations can be changed

Note 1: Register values can only be changed if PPSCON<IOLOCK> = 0.

REGISTER 10-18: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22 (BANKED EFCh)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable if IOLOCK = 0		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **SCK2R<4:0>:** Assign SPI2 Clock Input (SCLK2) to the Corresponding RPn Pin bits

REGISTER 10-19: RPINR23: PERIPHERAL PIN SELECT INPUT REGISTER 23 (BANKED EFDh)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable if IOLOCK = 0		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **SS2R<4:0>:** Assign SPI2 Slave Select Input (SS2IN) to the Corresponding RPn Pin bits

REGISTER 10-20: RPINR24: PERIPHERAL PIN SELECT INPUT REGISTER 24 (BANKED EFEh)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable if IOLOCK = 0		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **OCFAR<4:0>:** Assign PWM Fault Input (FLT0) to the Corresponding RPn Pin bits

© 2011 Microchip Technology Inc.

11.0 PARALLEL MASTER PORT (PMP)

The Parallel Master Port module (PMP) is an 8-bit parallel I/O module, specifically designed to communicate with a wide variety of parallel devices, such as communication peripherals, LCDs, external memory devices and microcontrollers. Because the interface to parallel peripherals varies significantly, the PMP is highly configurable. The PMP module can be configured to serve as either a PMP or as a Parallel Slave Port (PSP).

Key features of the PMP module are:

- Up to 16 bits of Addressing when Using Data/Address Multiplexing
- Up to 8 Programmable Address Lines
- One Chip Select Line
- Programmable Strobe Options:
  - Individual Read and Write Strobes or;
  - Read/$\overline{\text{Write}}$ Strobe with Enable Strobe
- Address Auto-Increment/Auto-Decrement
- Programmable Address/Data Multiplexing
- Programmable Polarity on Control Signals
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support:
  - Address Support
  - 4-Byte Deep, Auto-Incrementing Buffer
- Programmable Wait States
- Selectable Input Voltage Levels

FIGURE 11-1: PMP MODULE OVERVIEW

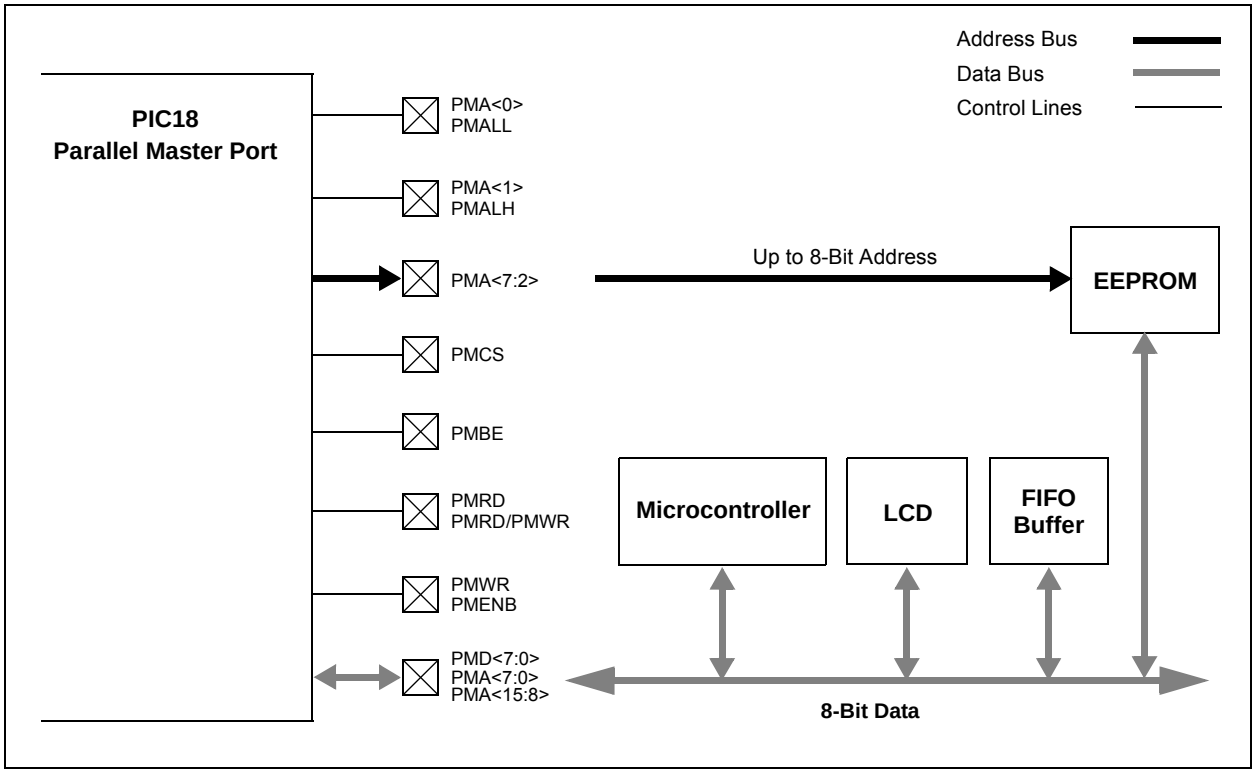


TABLE 17-3: RTCVALH AND RTCVALL REGISTER MAPPING

RTCPTR<1:0>	RTCC Value Register Window	
	RTCVALH<15:8>	RTCVALL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALRMCFG<1:0>) to select the desired Alarm register pair.

By reading or writing to the ALRMVALH register, the Alarm Pointer value, ALRMPTR<1:0>, decrements by 1 until it reaches '00'. Once it reaches '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 17-4: ALRMVAL REGISTER MAPPING

ALRMPTR<1:0>	Alarm Value Register Window	
	ALRMVALH<15:8>	ALRMVALL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

17.2.9 CALIBRATION

The real-time crystal input can be calibrated using the periodic auto-adjust feature. When properly calibrated, the RTCC can provide an error of less than three seconds per month.

To perform this calibration, find the number of error clock pulses and store the value in the lower half of the RTCCAL register. The 8-bit, signed value – loaded into RTCCAL – is multiplied by '4' and will either be added or subtracted from the RTCC timer, once every minute.

To calibrate the RTCC module:

1. Use another timer resource on the device to find the error of the 32.768 kHz crystal.
2. Convert the number of error clock pulses per minute (see Equation 17-1).

EQUATION 17-1: CONVERTING ERROR CLOCK PULSES

(Ideal Frequency (32,768) – Measured Frequency) * 60 = Error Clocks per Minute

- If the oscillator is *faster* than ideal (negative result from step 2), the RTCCALL register value needs to be negative. This causes the specified number of clock pulses to be subtracted from the timer counter once every minute.
- If the oscillator is *slower* than ideal (positive result from step 2), the RTCCALL register value needs to be positive. This causes the specified number of clock pulses to be added to the timer counter once every minute.

3. Load the RTCCAL register with the correct value.

Writes to the RTCCAL register should occur only when the timer is turned off, or immediately after the rising edge of the seconds pulse.

Note: In determining the crystal's error value, it is the user's responsibility to include the crystal's initial error from drift due to temperature or crystal aging.

© 2011 Microchip Technology Inc.

FIGURE 19-23: I²C™ MASTER MODE WAVEFORM (TRANSMISSION, 7-BIT OR 10-BIT ADDRESS)

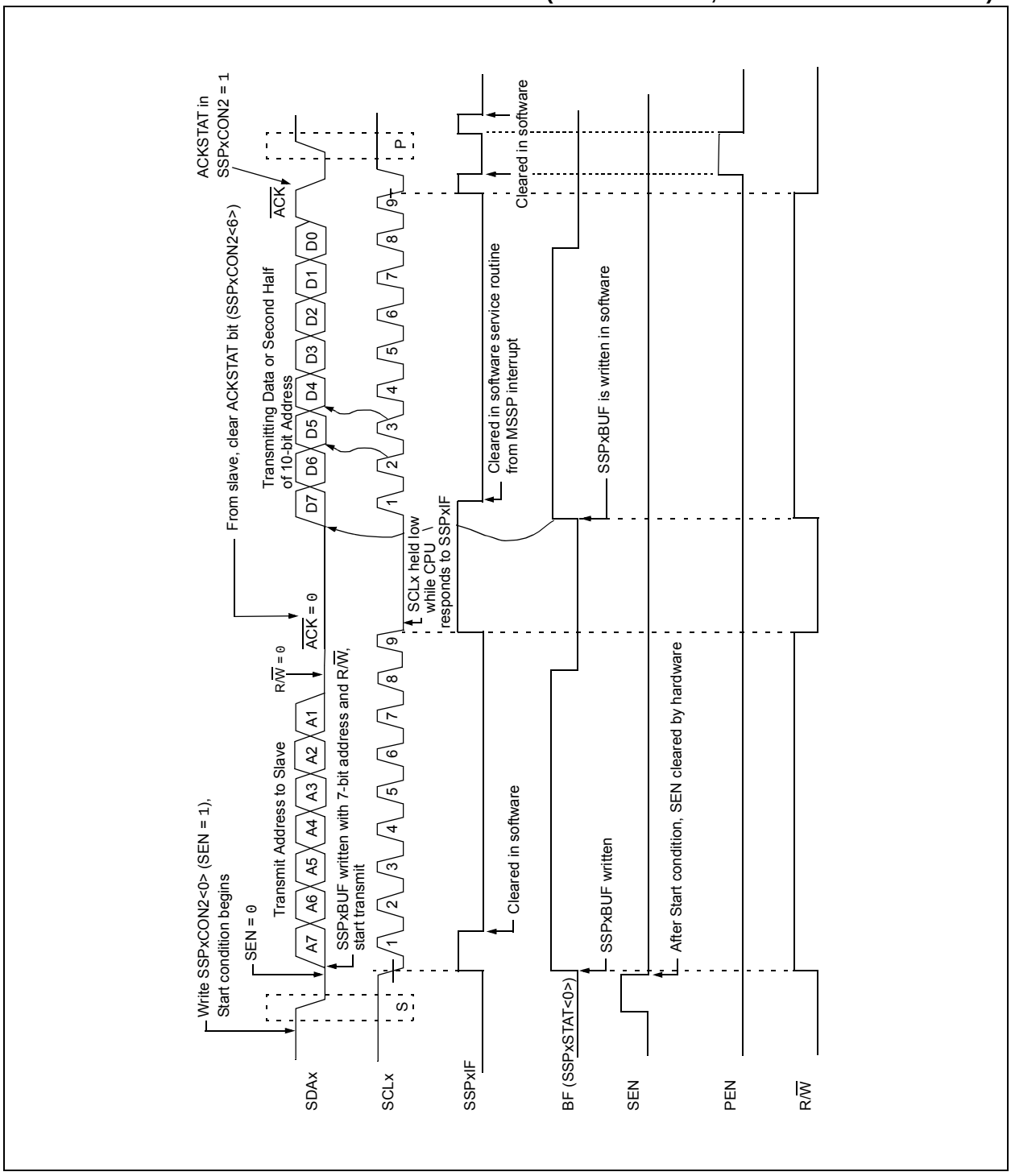


TABLE 21-2: SUMMARY OF A/D REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF(1)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE(1)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP(1)	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	72
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	72
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	72
ADRESH	A/D Result Register High Byte								70
ADRESL	A/D Result Register Low Byte								70
ADCON0	VCFG1	VCFG0	CHS3	CHS3	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	70
ANCON0	PCFG7(1)	PCFG6(1)	PCFG5(1)	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	74
ADCON1	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	70
ANCON1	VBGEN	r(2)	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	74
CCPxCON	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	71
PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	72
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72

Legend: — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

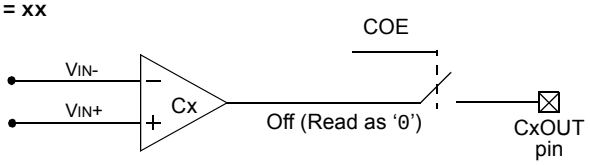
Note 1: These bits are only available on 44-pin devices.

2: Reserved. Always maintain as '0' for minimum power consumption.

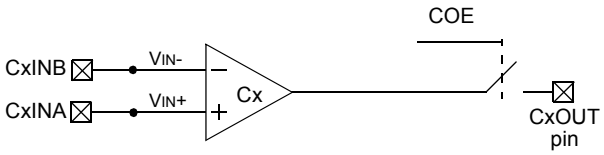
© 2011 Microchip Technology Inc.

FIGURE 22-4: COMPARATOR CONFIGURATIONS

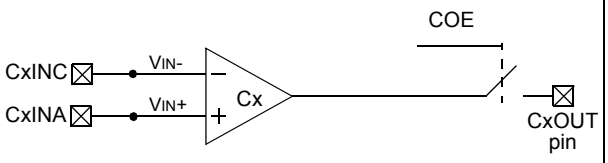
Comparator Off
CON = 0, CREF = x, CCH<1:0> = xx



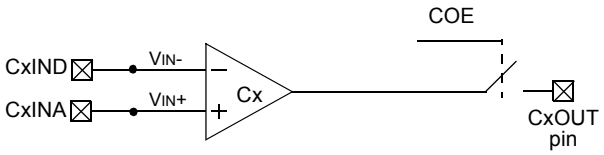
Comparator CxINB > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 00



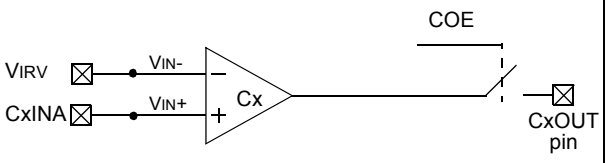
Comparator CxINC > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 01



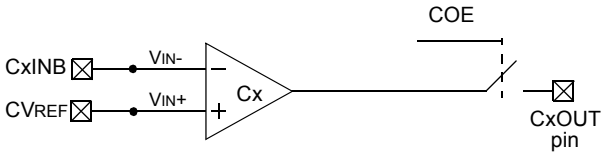
Comparator CxIND > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 10



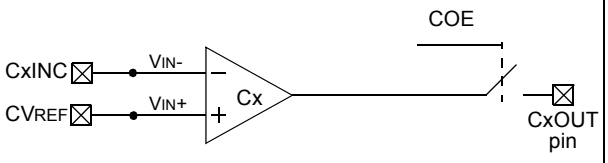
Comparator VIRV > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 11



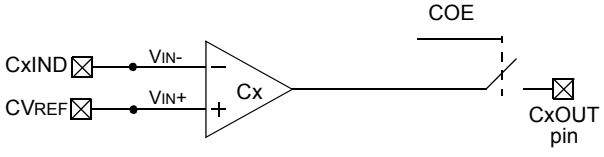
Comparator CxINB > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 00



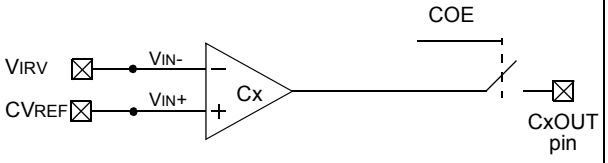
Comparator CxINC > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 01



Comparator CxIND > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 10



Comparator VIRV > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 11



Note: VIRV is the Internal Reference Voltage (see Table 29-2).

© 2011 Microchip Technology Inc.

22.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from Sleep mode when enabled. Each operational comparator will consume additional current. To minimize power consumption while in Sleep mode, turn off the comparators (CON = 0) before entering Sleep. If the device wakes up from Sleep, the contents of the CMxCON register are not affected.

22.8 Effects of a Reset

A device Reset forces the CMxCON registers to their Reset state. This forces both comparators and the voltage reference to the OFF state.

TABLE 22-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

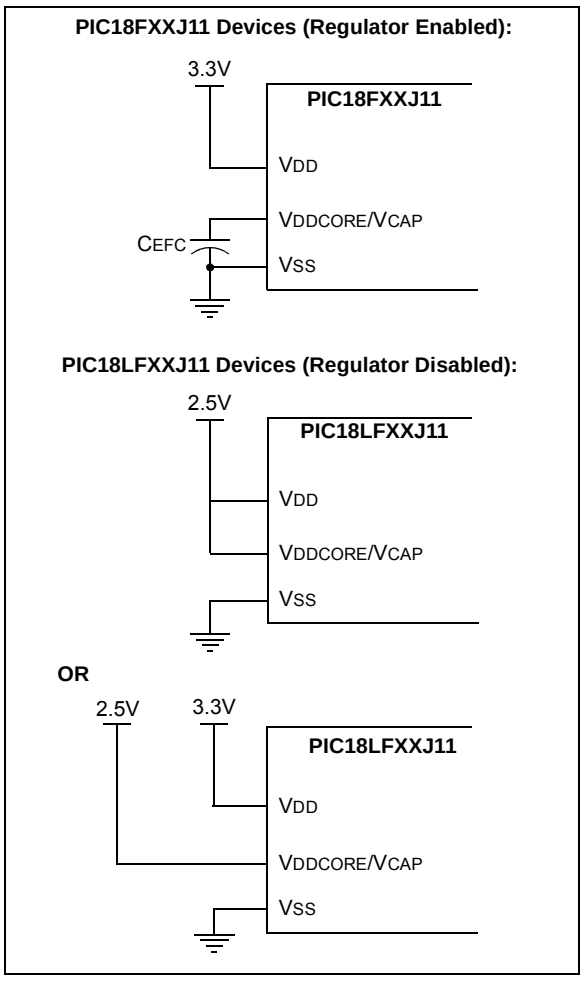
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	72
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	72
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	72
CMxCON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	70
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	74
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	73
ANCON0	PCFG7(1)	PCFG6(1)	PCFG5(1)	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	74
PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	72
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72

Legend: — = unimplemented, read as '0', r = reserved. Shaded cells are not related to comparator operation.

Note 1: These bits and/or registers are not implemented on 28-pin devices.

© 2011 Microchip Technology Inc.

FIGURE 26-2: CONNECTIONS FOR THE ON-CHIP REGULATOR



26.3.2 ON-CHIP REGULATOR AND BOR

When the on-chip regulator is enabled, PIC18F46J11 family devices also have a simple brown-out capability. If the voltage supplied to the regulator is inadequate to maintain a minimum output level; the regulator Reset circuitry will generate a Brown-out Reset (BOR). This event is captured by the $\overline{\text{BOR}}$ flag bit (RCON<0>).

The operation of the BOR is described in more detail in **Section 5.4 "Brown-out Reset (BOR)"** and **Section 5.4.1 "Detecting BOR"**. The brown-out voltage levels are specific in **Section 29.1 "DC Characteristics: Supply Voltage PIC18F46J11 Family (Industrial)"**.

26.3.3 POWER-UP REQUIREMENTS

The on-chip regulator is designed to meet the power-up requirements for the device. If the application does not use the regulator, then strict power-up conditions must be adhered to. While powering up, V_{DDCORE} should not exceed V_{DD} by 0.3 volts.

26.3.4 OPERATION IN SLEEP MODE

When enabled, the on-chip regulator always consumes a small incremental amount of current over I_{DD}. This includes when the device is in Sleep mode, even though the core digital logic does not require much power. To provide additional savings in applications where power resources are critical, the regulator can be configured to automatically enter a lower quiescent draw standby mode whenever the device goes into Sleep mode. This feature is controlled by the REGSLP bit (WDTCON<7>, Register 26-11). If this bit is set upon entry into Sleep mode, the regulator will transition into a lower power state. In this state, the regulator still provides a regulated output voltage necessary to maintain SRAM state information, but consumes less quiescent current.

Substantial Sleep mode power savings can be obtained by setting the REGSLP bit, but device wake-up time will increase in order to insure the regulator has enough time to stabilize.

© 2011 Microchip Technology Inc.

BTG **Bit Toggle f**

Syntax: BTG f, b {,a}

Operands: $0 \le f \le 255$
$0 \le b < 7$
$a \in [0,1]$

Operation: $(\overline{f<b>}) \rightarrow f<b>$

Status Affected: None

Encoding:

0111	bbba	ffff	ffff

Description: Bit ‘b’ in data memory location ‘f’ is inverted.

If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank (default).

If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 27.2.3 “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 register ‘f’

Example: `BTG LATC, 4, 0`

Before Instruction:
LATC = 0111 0101 [75h]
After Instruction:
LATC = 0110 0101 [65h]

BOV **Branch if Overflow**

Syntax: BOV n

Operands: $-128 \le n \le 127$

Operation: if Overflow bit is ‘1’,
$(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0100	nnnn	nnnn

Description: If the Overflow bit is ‘1’, then the program will branch.

The 2’s complement number ‘2n’ is added 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 then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal ‘n’	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal ‘n’	Process Data	No operation

Example:
```
HERE    BOV  Jump
```
Before Instruction
PC = address (HERE)
After Instruction
If Overflow = 1;
PC = address (Jump)
If Overflow = 0;
PC = address (HERE + 2)

BZ **Branch if Zero**

Syntax: BZ n

Operands: -128 ≤ n ≤ 127

Operation: if Zero bit is '1',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0000	nnnn	nnnn

Description: If the Zero bit is '1', then the program will branch.

The 2's complement number '2n' is added 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 then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

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

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BZ Jump

Before Instruction
PC = address (HERE)
After Instruction
If Zero = 1;
PC = address (Jump)
If Zero = 0;
PC = address (HERE + 2)

CALL **Subroutine Call**

Syntax: CALL k {,s}

Operands: 0 ≤ k ≤ 1048575
s ∈ [0,1]

Operation: (PC) + 4 → TOS,
k → PC<20:1>;
if s = 1,
(W) → WS,
(STATUS) → STATUSS,
(BSR) → BSRS

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	110s	$k_7$kkk	kkkk$_0$
2nd word(k<19:8>)	1111	k_{19}kkk	kkkk	kkkk$_8$

Description: Subroutine call of entire 2-Mbyte memory range. First, return address (PC + 4) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR
registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into PC<20:1>. CALL is a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: HERE CALL THERE,1

Before Instruction
PC = address (HERE)
After Instruction
PC = address (THERE)
TOS = address (HERE + 4)
WS = W
BSRS = BSR
STATUSS = STATUS

© 2011 Microchip Technology Inc.

POP **Pop Top of Return Stack**

Syntax: POP

Operands: None

Operation: (TOS) → bit bucket

Status Affected: None

Encoding:

0000	0000	0000	0110

Description: The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	POP TOS value	No operation

Example:
```
POP
GOTO NEW
```
Before Instruction
TOS = 0031A2h
Stack (1 level down) = 014332h

After Instruction
TOS = 014332h
PC = NEW

PUSH **Push Top of Return Stack**

Syntax: PUSH

Operands: None

Operation: (PC + 2) → TOS

Status Affected: None

Encoding:

0000	0000	0000	0101

Description: The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	PUSH PC + 2 onto return stack	No operation	No operation

Example:
```
PUSH
```
Before Instruction
TOS = 345Ah
PC = 0124h

After Instruction
PC = 0126h
TOS = 0126h
Stack (1 level down) = 345Ah

© 2011 Microchip Technology Inc.

SUBFSR **Subtract Literal from FSR**

Syntax: SUBFSR f, k

Operands: $0 \le k \le 63$

$f \in [0, 1, 2]$

Operation: FSRf – k → FSRf

Status Affected: None

Encoding:

1110	1001	ffkk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `SUBFSR 2, 0x23`

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 03DCh

SUBULNK **Subtract Literal from FSR2 and Return**

Syntax: SUBULNK k

Operands: $0 \le k \le 63$

Operation: FSR2 – k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:

1110	1001	11kk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR2. A `RETURN` is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a `NOP` is performed during the second cycle.

This may be thought of as a special case of the `SUBFSR` instruction, where f = 3 (binary '`11`'); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination
No Operation	No Operation	No Operation	No Operation

Example: `SUBULNK 0x23`

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 03DCh
PC = (TOS)

FIGURE 29-13: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)

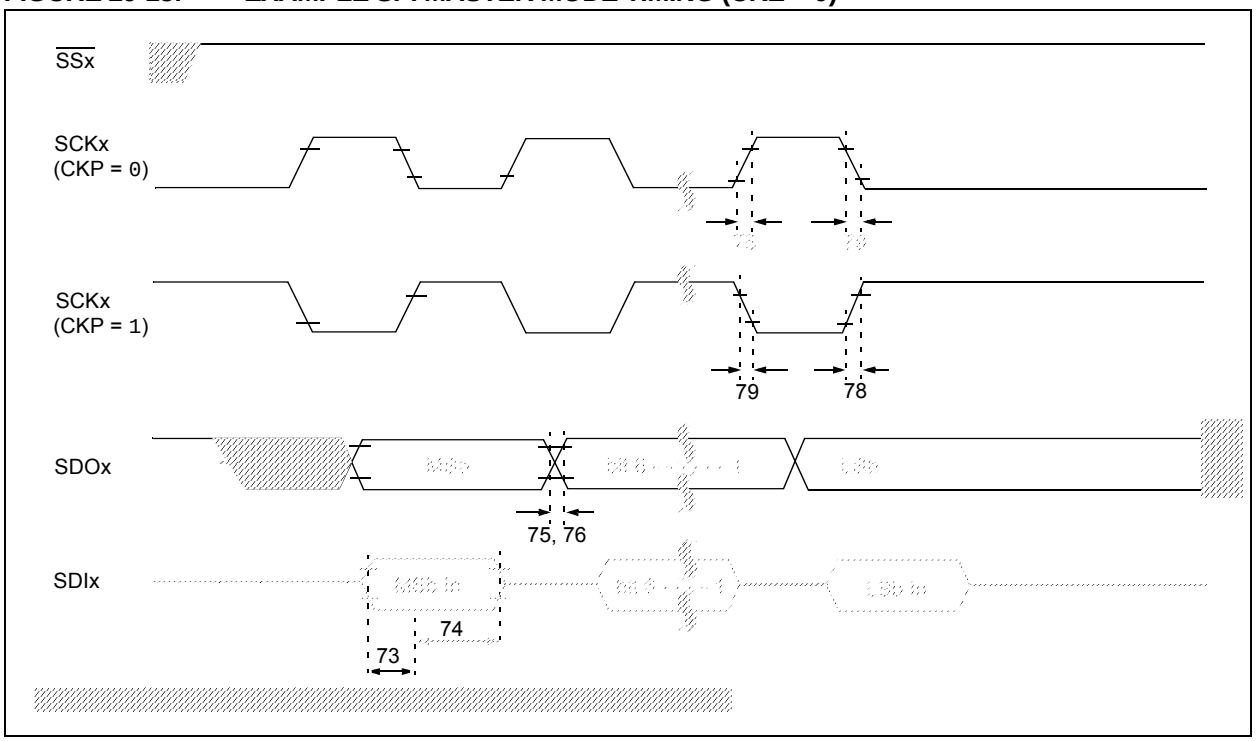
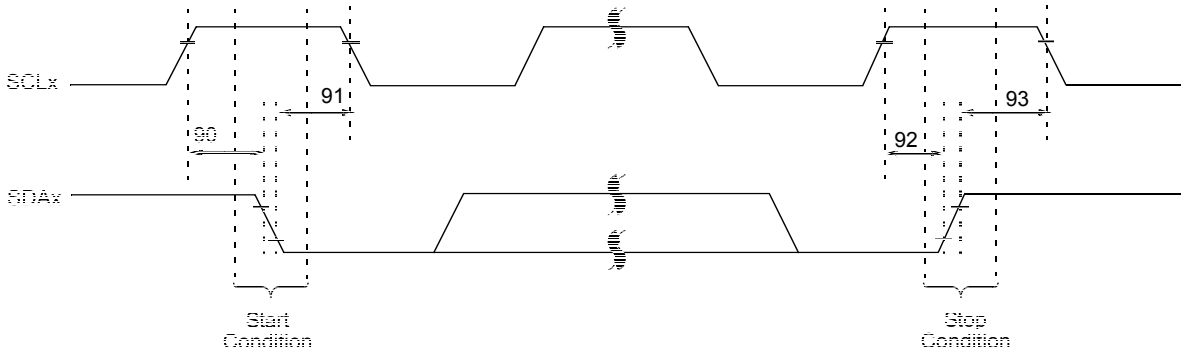


TABLE 29-20: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TDIV2SCH, TDIV2SCL	Setup Time of SDIx Data Input to SCKx Edge	35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			100	—	ns	VDD = 2.15V, VDDCORE = 2.15V
74	TSCH2DIL, TSCL2DIL	Hold Time of SDIx Data Input to SCKx Edge	30	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			83	—	ns	VDD = 2.15V
75	TDOR	SDOx Data Output Rise Time	—	25	ns	PORTB or PORTC
76	TDOF	SDOx Data Output Fall Time	—	25	ns	PORTB or PORTC
78	TSCR	SCKx Output Rise Time (Master mode)	—	25	ns	PORTB or PORTC
79	TSCF	SCKx Output Fall Time (Master mode)	—	25	ns	PORTB or PORTC

© 2011 Microchip Technology Inc.

FIGURE 29-17: I²C™ BUS START/STOP BITS TIMING

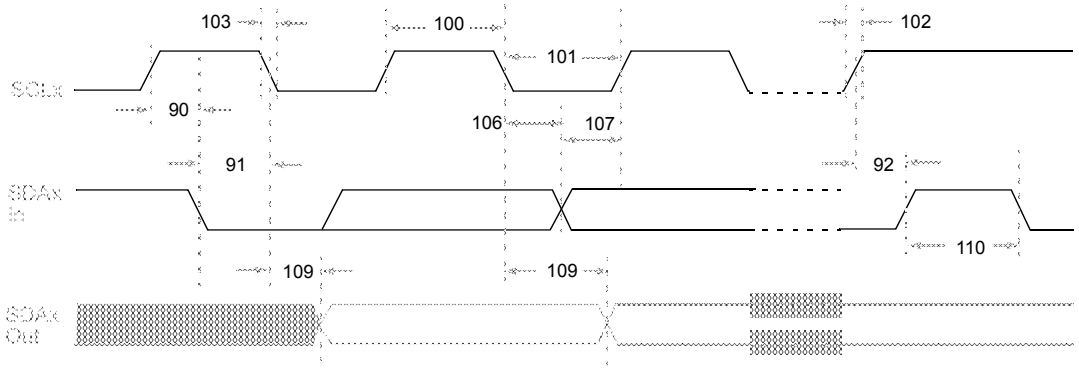


Note: Refer to Figure 29-4 for load conditions.

TABLE 29-24: I²C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4700	—	ns	
			400 kHz mode	600	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—		

FIGURE 29-18: I²C™ BUS DATA TIMING



Note: Refer to Figure 29-4 for load conditions.

© 2011 Microchip Technology Inc.