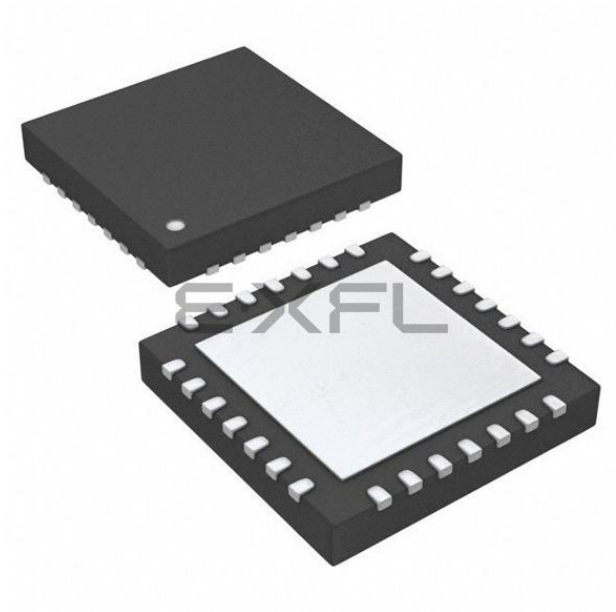


Welcome to E-XFL.COM

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	25MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1320-e-ml

3.5.2 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock (defined in Configuration Register 1H) becomes ready. At that time, the OSTS bit is set and the device begins executing code.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see **Section 19.3 "Two-Speed Start-up"**) or Fail-Safe Clock Monitor (see **Section 19.4 "Fail-Safe Clock Monitor"**) are enabled in Configuration Register 1H, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Since the OSCCON register is cleared following all Resets, the INTRC clock source is selected. A higher speed clock may be selected by modifying the IRCF bits in the OSCCON register. Execution is clocked by the internal oscillator block until either the primary clock becomes ready, or a power managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

3.5.3 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions, depending on which power managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in a wake from the power managed mode (see Sections 3.2 through 3.4).

If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 19.2 "Watchdog Timer (WDT)"**).

The WDT timer and postscaler are cleared by executing a `SLEEP` or `CLRWDT` instruction, the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifying the IRCF bits in the OSCCON register if the internal oscillator block is the system clock source.

3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power managed modes do not invoke the OST at all. These are:

- PRI_IDLE mode, where the primary clock source is not stopped; or
- the primary clock source is not any of LP, XT, HS or HSPLL modes.

In these cases, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes).

However, a fixed delay (approximately 10 μs) following the wake event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

3.6 INTOSC Frequency Drift

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz (see Table 22-6). However, this frequency may drift as V_{DD} or temperature changes, which can affect the controller operation in a variety of ways.

It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register (Register 2-1). This has the side effect that the INTRC clock source frequency is also affected. However, the features that use the INTRC source often do not require an exact frequency. These features include the Fail-Safe Clock Monitor, the Watchdog Timer and the RC_RUN/RC_IDLE modes when the INTRC clock source is selected.

Being able to adjust the INTOSC requires knowing when an adjustment is required, in which direction it should be made and in some cases, how large a change is needed. Three examples follow but other techniques may be used.

TABLE 4-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
BSR	1220	1320	---- 0000	---- 0000	---- uuuu
INDF2	1220	1320	N/A	N/A	N/A
POSTINC2	1220	1320	N/A	N/A	N/A
POSTDEC2	1220	1320	N/A	N/A	N/A
PREINC2	1220	1320	N/A	N/A	N/A
PLUSW2	1220	1320	N/A	N/A	N/A
FSR2H	1220	1320	---- 0000	---- 0000	---- uuuu
FSR2L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	1220	1320	---x xxxx	---u uuuu	---u uuuu
TMR0H	1220	1320	0000 0000	0000 0000	uuuu uuuu
TMR0L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	1220	1320	1111 1111	1111 1111	uuuu uuuu
OSCCON	1220	1320	0000 q000	0000 q000	uuuu qquu
LVDCON	1220	1320	--00 0101	--00 0101	--uu uuuu
WDTCON	1220	1320	---- ---0	---- ---0	---- ---u
RCON(4)	1220	1320	0--1 11q0	0--q qquu	u--u qquu
TMR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	1220	1320	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	1220	1320	0000 0000	0000 0000	uuuu uuuu
PR2	1220	1320	1111 1111	1111 1111	1111 1111
T2CON	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADRESH	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1220	1320	00-0 0000	00-0 0000	uu-u uuuu
ADCON1	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADCON2	1220	1320	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
PWM1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
ECCPAS	1220	1320	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 4-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the Oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: Bit 5 of PORTA is enabled if $\overline{\text{MCLR}}$ is disabled.

5.13 Status Register

The STATUS register, shown in Register 5-2, contains the arithmetic status of the ALU. As with any other SFR, it can be the operand for any instruction.

If the STATUS register is the destination for an instruction that affects the Z, DC, C, OV or N bits, the results of the instruction are not written; instead, the status is updated according to the instruction performed. Therefore, the result of an instruction with the STATUS register as its destination may be different than intended. As an example, CLRF STATUS will set the Z bit and leave the remaining Status bits unchanged ('000u u1uu').

It is recommended that only BCF, BSF, SWAPF, MOVFF and MOVWF instructions are used to alter the STATUS register, because these instructions do not affect the Z, C, DC, OV or N bits in the STATUS register.

For other instructions that do not affect Status bits, see the instruction set summaries in Table 20-1.

Note: The C and DC bits operate as the borrow and digit borrow bits, respectively, in subtraction.

REGISTER 5-2: STATUS: STATUS REGISTER

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	N	OV	Z	DC	C
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 condition

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

bit 4 **N:** Negative bit
This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).
1 = Result was negative
0 = Result was positive

bit 3 **OV:** Overflow bit
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit 7) to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/Digit Borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)(1)
1 = A carry-out from the 4th low-order bit of the result occurred
0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/Borrow bit(1) (ADDWF, ADDLW, SUBLW, SUBWF instructions)(1)
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note 1: For Borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high-order or low-order bit of the source register.

REGISTER 6-1: EECON1: EEPROM CONTROL 1 REGISTER

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR(1)	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	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 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
- 1 = Access program Flash memory
- 0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
- 1 = Accesses Configuration, User ID and Device ID Registers
- 0 = Accesses Flash Program or data EEPROM Memory

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **FREE:** Flash Row Erase Enable bit
- 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation – TBLPTR<5:0> are ignored)
- 0 = Perform write only

bit 3 **WRERR:** EEPROM Error Flag bit(1)
- 1 = A write operation was prematurely terminated (any Reset during self-timed programming)
- 0 = The write operation completed normally

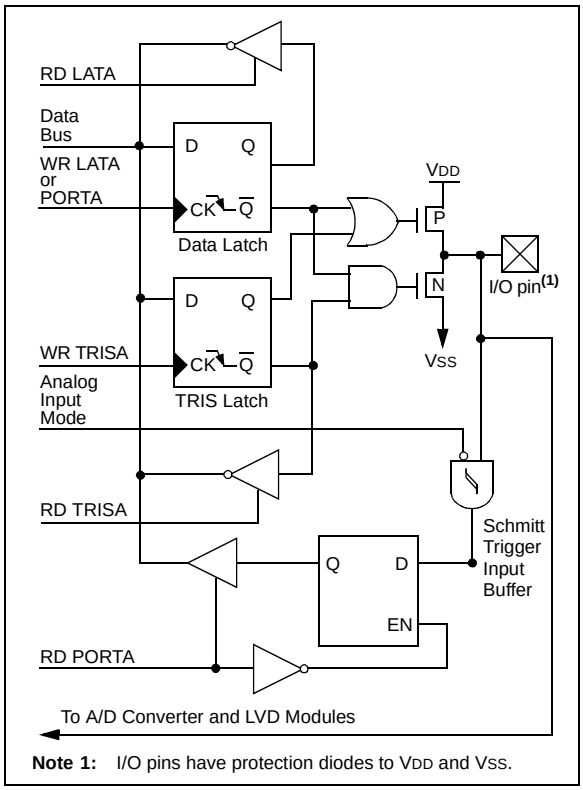
bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
- 0 = Write cycle completed

bit 0 **RD:** Read Control bit
- 1 = Initiates a memory read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
- 0 = Read completed

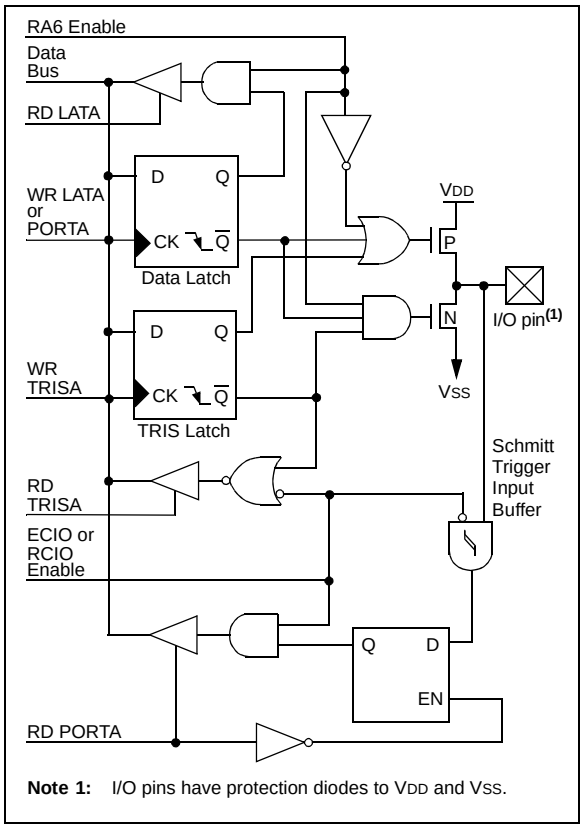
Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

FIGURE 10-2: BLOCK DIAGRAM OF RA3:RA0 PINS



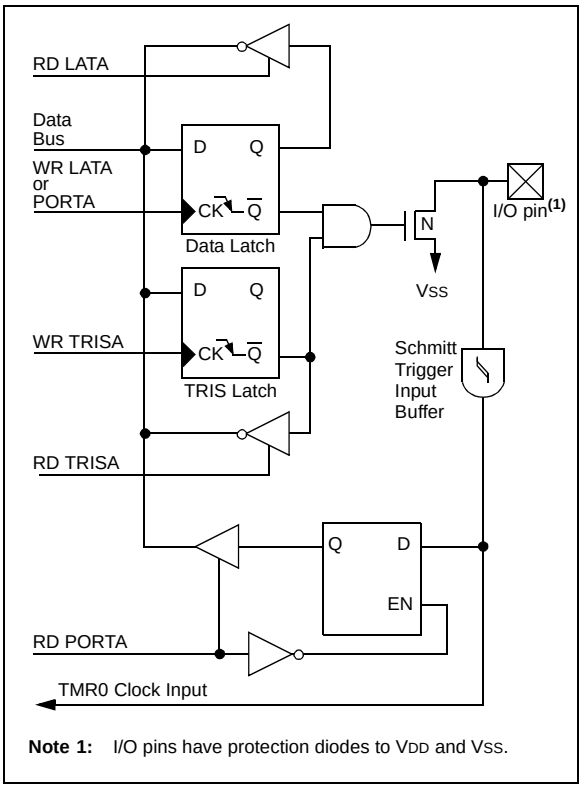
Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-3: BLOCK DIAGRAM OF OSC2/CLKO/RA6 PIN



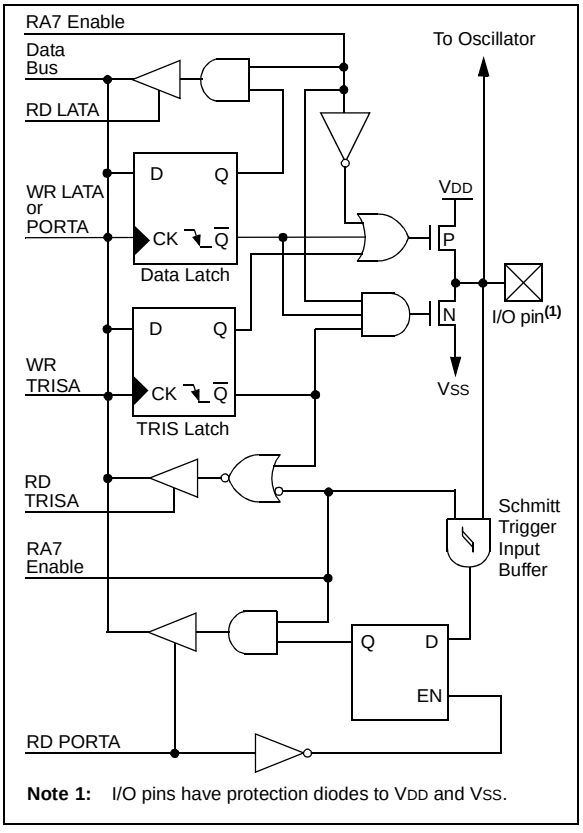
Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-4: BLOCK DIAGRAM OF RA4/T0CKI PIN



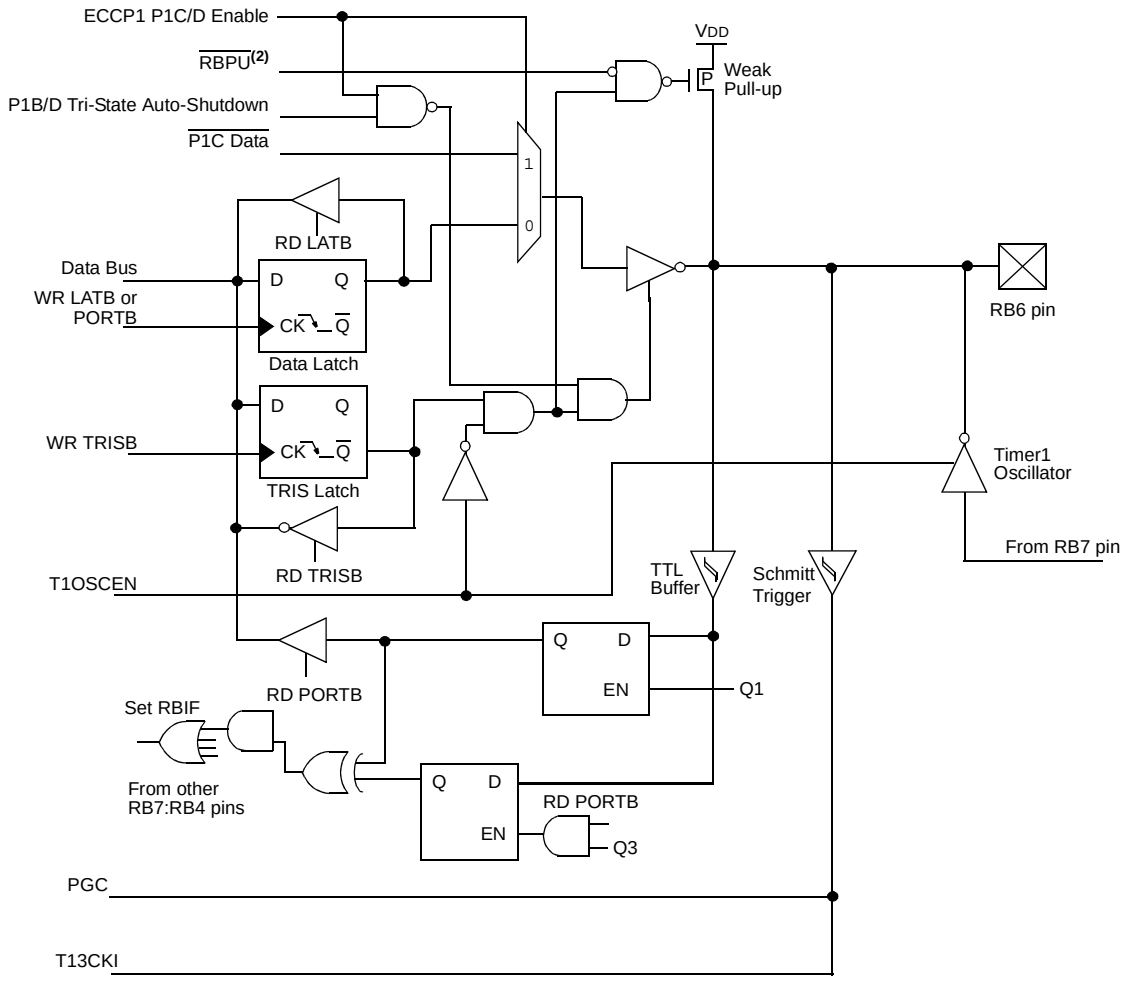
Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-5: BLOCK DIAGRAM OF OSC1/CLKI/RA7 PIN



Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-13: BLOCK DIAGRAM OF RB6/PGC/T1OSO/T13CKI/P1C/KBI2 PIN



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (INTCON2<7>).

14.0 TIMER3 MODULE

The Timer3 module timer/counter has the following features:

- 16-bit timer/counter
  (two 8-bit registers; TMR3H and TMR3L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt-on-overflow from FFFFh to 0000h
- Reset from CCP module trigger

Figure 14-1 is a simplified block diagram of the Timer3 module.

Register 14-1 shows the Timer3 Control register. This register controls the operating mode of the Timer3 module and sets the CCP clock source.

Register 12-1 shows the Timer1 Control register. This register controls the operating mode of the Timer1 module, as well as contains the Timer1 Oscillator Enable bit (T1OSCEN), which can be a clock source for Timer3.

REGISTER 14-1: T3CON: TIMER3 CONTROL REGISTER

R/W-0/0	U-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
RD16	—	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
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 **RD16:** 16-bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6 **Unimplemented**: Read as ‘0’

bit 5-4 **T3CKPS<1:0>:** Timer3 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T3CCP1:** Timer3 and Timer1 to CCP1 Enable bits
1 = Timer3 is the clock source for compare/capture CCP module
0 = Timer1 is the clock source for compare/capture CCP module

bit 2 **$\overline{\text{T3SYNC}}$:** Timer3 External Clock Input Synchronization Control bit
(Not usable if the system clock comes from Timer1/Timer3.)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI
(on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

REGISTER 16-1: TXSTA: TRANSMIT STATUS AND 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-1/1	R/W-0/0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don't care.

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit

Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed

Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR Idle
0 = TSR busy

bit 0 **TX9D:** 9th bit of Transmit Data
Can be address/data bit or a parity bit.

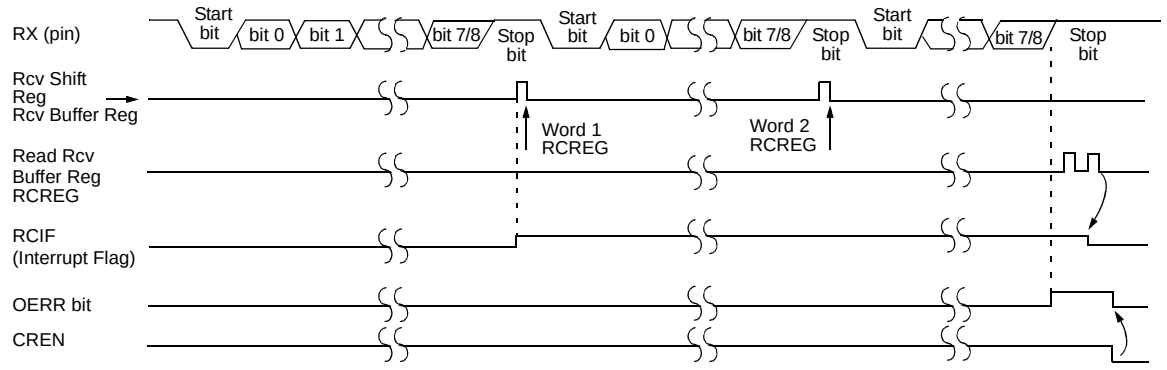
Note 1: SREN/CREN overrides TXEN in Sync mode.

To set up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (see **Section 16.2 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 16-6: ASYNCHRONOUS RECEPTION



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

TABLE 16-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
RCREG	EUSART Receive Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	Baud Rate Generator Register High Byte								0000 0000	0000 0000
SPBRG	Baud Rate Generator Register Low Byte								0000 0000	0000 0000

Legend: x = unknown, – = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (AV_{DD} and AV_{SS}), or the voltage level on the RA3/AN3/V_{REF+} and RA2/AN2/V_{REF-} pins.

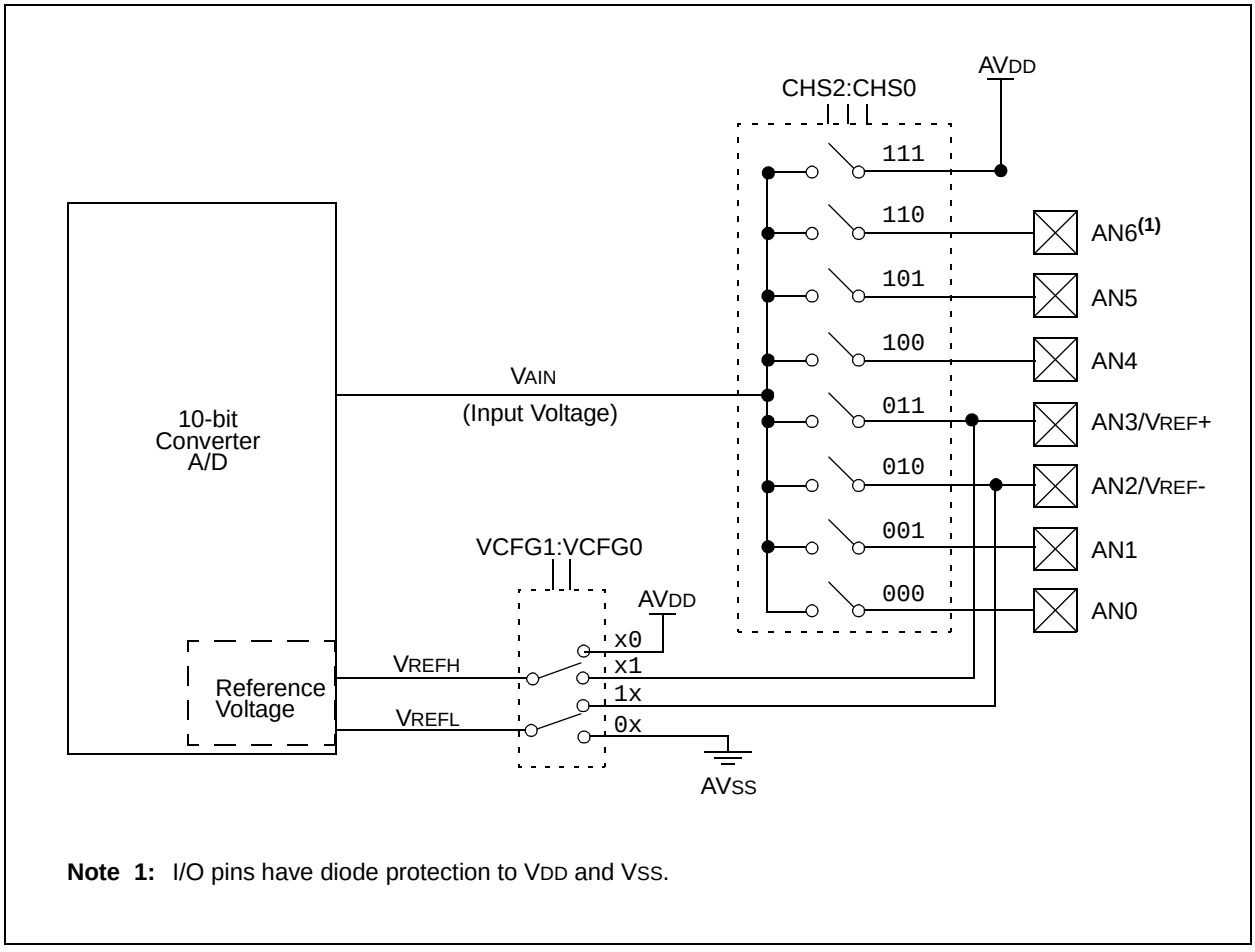
The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted.

Each port pin associated with the A/D converter can be configured as an analog input, or as a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH/ADRESL registers, the GO/$\overline{DONE}$ bit (ADCON0 register) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 17-1.

FIGURE 17-1: A/D BLOCK DIAGRAM



Note 1: I/O pins have diode protection to V_{DD} and V_{SS}.

17.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 17-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 17-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Example 17-1 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	120 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	5V → RSS = 7 kΩ
Temperature	=	50°C (system max.)
VHOLD	=	0V @ time = 0

17.2 A/D VREF+ and VREF- References

If external voltage references are used instead of the internal AVDD and AVSS sources, the source impedance of the VREF+ and VREF- voltage sources must be considered. During acquisition, currents supplied by these sources are insignificant. However, during conversion, the A/D module sinks and sources current through the reference sources.

In order to maintain the A/D accuracy, the voltage reference source impedances should be kept low to reduce voltage changes. These voltage changes occur as reference currents flow through the reference source impedance. **The maximum recommended impedance of the VREF+ and VREF- external reference voltage sources is 250Ω.**

EQUATION 17-1: ACQUISITION TIME

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$

EQUATION 17-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (\Delta V_{REF} - (\Delta V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$
or
$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2048)$$

EXAMPLE 17-1: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$
$T_{AMP} = 5\ \mu s$
$T_{COFF} = (Temp - 25^oC)(0.05\ \mu s/^oC)$
$(50^oC - 25^oC)(0.05\ \mu s/^oC)$
$1.25\ \mu s$
Temperature coefficient is only required for temperatures > 25ºC. Below 25ºC, $T_{COFF} = 0\ \mu s$.
$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2047)\ \mu s$
$-(120\ pF)(1\ k\Omega + 7\ k\Omega + 2.5\ k\Omega) \ln(0.0004883)\ \mu s$
$9.61\ \mu s$
$T_{ACQ} = 5\ \mu s + 1.25\ \mu s + 9.61\ \mu s$
$12.86\ \mu s$

REGISTER 19-4: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

R/P-1	U-0	U-0	U-0	U-0	U-0	U-0	U-0
MCLRE	—	—	—	—	—	—	—
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	P = Programmable bit

bit 7 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit
1 = $\overline{\text{MCLR}}$ pin enabled, RA5 input pin disabled
0 = RA5 input pin enabled, $\overline{\text{MCLR}}$ disabled

bit 6-0 **Unimplemented:** Read as '0'

REGISTER 19-5: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVR
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	P = Programmable bit

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit (see note)
1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins
0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug

bit 6-3 **Unimplemented:** Read as '0'

bit 2 **LVP:** Low-Voltage ICSP Enable bit
1 = Low-Voltage ICSP enabled
0 = Low-Voltage ICSP disabled

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

bit 0 **STVR:** Stack Full/Underflow Reset Enable bit
1 = Stack full/underflow will cause Reset
0 = Stack full/underflow will not cause Reset

Note: The Timer1 oscillator shares the T1OSI and T1OSO pins with the PGD and PGC pins used for programming and debugging.

When using the Timer1 oscillator, In-Circuit Serial Programming (ICSP) may not function correctly (high voltage or low voltage), or the In-Circuit Debugger (ICD) may not communicate with the controller. As a result of using either ICSP or ICD, the Timer1 crystal may be damaged.

If ICSP or ICD operations are required, the crystal should be disconnected from the circuit (disconnect either lead) or installed after programming. The oscillator loading capacitors may remain in-circuit during ICSP or ICD operation.

BNOV **Branch if Not Overflow**

Syntax: [*label*] BNOV n

Operands: -128 ≤ n ≤ 127

Operation: if Overflow bit is '0'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0101	nnnn	nnnn

Description: If the Overflow bit is '0', 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 2-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 BNOV Jump

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

BNZ **Branch if Not Zero**

Syntax: [*label*] BNZ n

Operands: -128 ≤ n ≤ 127

Operation: if Zero bit is '0'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0001	nnnn	nnnn

Description: If the Zero bit is '0', 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 2-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 BNZ Jump

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

POP **Pop Top of Return Stack**

Syntax: [*label*] 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 = 0x0031A2
Stack (1 level down) = 0x014332

After Instruction
TOS = 0x014332
PC = NEW

PUSH **Push Top of Return Stack**

Syntax: [*label*] 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 = 0x00345A
PC = 0x000124

After Instruction
PC = 0x000126
TOS = 0x000126
Stack (1 level down) = 0x00345A

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: $k - (W) \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1000	kkkk	kkkk

Description: W is subtracted from the 8-bit literal ‘k’. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example 1: `SUBLW 0x02`

Before Instruction
W = 1
C = ?

After Instruction
W = 1
C = 1 ; result is positive
Z = 0
N = 0

Example 2: `SUBLW 0x02`

Before Instruction
W = 2
C = ?

After Instruction
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: `SUBLW 0x02`

Before Instruction
W = 3
C = ?

After Instruction
W = FF ; (2’s complement)
C = 0 ; result is negative
Z = 0
N = 1

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: $(f) - (W) \rightarrow dest$

Status Affected: N, OV, C, DC, Z

Encoding:

0101	11da	ffff	ffff

Description: Subtract W from register ‘f’ (2’s complement method). If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’ (default). If ‘a’ is ‘0’, the Access Bank will be selected, overriding the BSR value. If ‘a’ is ‘1’, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: `SUBWF REG`

Before Instruction
REG = 3
W = 2
C = ?

After Instruction
REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

Example 2: `SUBWF REG, W`

Before Instruction
REG = 2
W = 2
C = ?

After Instruction
REG = 2
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: `SUBWF REG`

Before Instruction
REG = 0x01
W = 0x02
C = ?

After Instruction
REG = 0xFFh ;(2’s complement)
W = 0x02
C = 0x00 ;result is negative
Z = 0x00
N = 0x01

21.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

21.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

TABLE 22-1: MEMORY PROGRAMMING REQUIREMENTS

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
		Internal Program Memory Programming Specifications(1)					
D110	VPP	Voltage on $\overline{MCLR}$/VPP pin	9.00	—	13.25	V	**(Note 2)**
D112	IPP	Current into $\overline{MCLR}$/VPP pin	—	—	5	μA	
D113	IDDP	Supply Current during Programming	—	—	10	mA	
		Data EEPROM Memory					
D120	ED	Byte Endurance	100K	1M	—	E/W	-40°C to +85°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	4	—	ms	
D123	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh(3)	1M	10M	—	E/W	-40°C to +85°C
		Program Flash Memory					
D130	EP	Cell Endurance	10K	100K	—	E/W	-40°C to +85°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VIE	VDD for Block Erase	4.5	—	5.5	V	Using ICSP port
D132A	VIW	VDD for Externally Timed Erase or Write	4.5	—	5.5	V	Using ICSP port
D132B	VPEW	VDD for Self-Timed Write	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D133	TIE	ICSP™ Block Erase Cycle Time	—	4	—	ms	VDD > 4.5V
D133A	TIW	ICSP Erase or Write Cycle Time (externally timed)	1	—	—	ms	VDD > 4.5V
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	
D134	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated

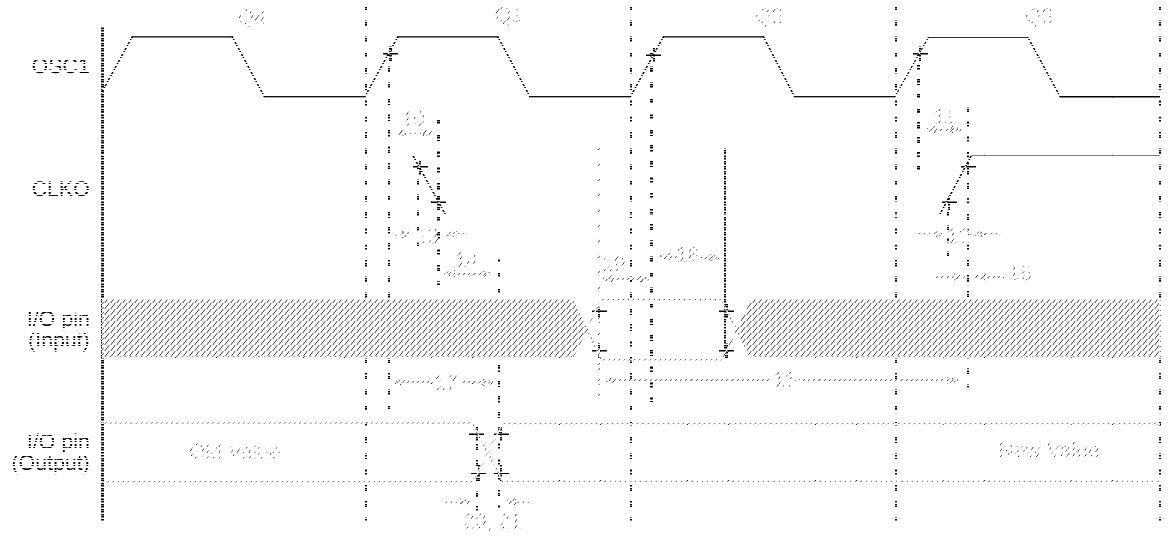
† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: These specifications are for programming the on-chip program memory through the use of table write instructions.

2: The pin may be kept in this range at times other than programming, but it is not recommended.

3: Refer to **Section 7.8 “Using the Data EEPROM”** for a more detailed discussion on data EEPROM endurance.

FIGURE 22-7: CLKO AND I/O TIMING



Note: Refer to Figure 22-5 for load conditions.

TABLE 22-7: CLKO AND I/O TIMING REQUIREMENTS

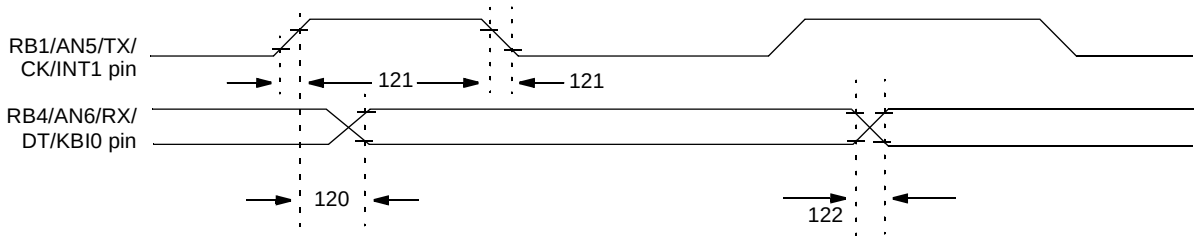
Param. No.	Symbol	Characteristic		Min.	Typ.	Max.	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKO↓		—	75	200	ns	**(Note 1)**
11	TosH2ckH	OSC1↑ to CLKO↑		—	75	200	ns	**(Note 1)**
12	TckR	CLKO Rise Time		—	35	100	ns	**(Note 1)**
13	TckF	CLKO Fall Time		—	35	100	ns	**(Note 1)**
14	TckL2ioV	CLKO↓ to Port Out Valid		—	—	0.5 TCY + 20	ns	**(Note 1)**
15	TioV2ckH	Port In Valid before CLKO↑		0.25 TCY + 25	—	—	ns	**(Note 1)**
16	TckH2ioI	Port In Hold after CLKO↑		0	—	—	ns	**(Note 1)**
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port Out Valid		—	50	150	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	PIC18F1X20	100	—	—	ns	
18A			PIC18LF1X20	200	—	—	ns	
19	TioV2osH	Port Input Valid to OSC1↑ (I/O in setup time)		0	—	—	ns	
20	TioR	Port Output Rise Time	PIC18F1X20	—	10	25	ns	
20A			PIC18LF1X20	—	—	60	ns	
21	TioF	Port Output Fall Time	PIC18F1X20	—	10	25	ns	
21A			PIC18LF1X20	—	—	60	ns	

Note 1: Measurements are taken in RC mode, where CLKO output is 4 x TOSC.

TABLE 22-10: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)

Param. No.	Symbol	Characteristic			Min.	Max.	Units	Conditions
50	TccL	CCPx Input Low Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18F1X20	10	—	ns	
				PIC18LF1X20	20	—	ns	
51	TccH	CCPx Input High Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18F1X20	10	—	ns	
				PIC18LF1X20	20	—	ns	
52	TccP	CCPx Input Period			3 TCY + 40 / N	—	ns	N = prescale value (1, 4 or 16)
53	TccR	CCPx Output Fall Time		PIC18F1X20	—	25	ns	
				PIC18LF1X20	—	45	ns	
54	TccF	CCPx Output Fall Time		PIC18F1X20	—	25	ns	
				PIC18LF1X20	—	45	ns	

FIGURE 22-12: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

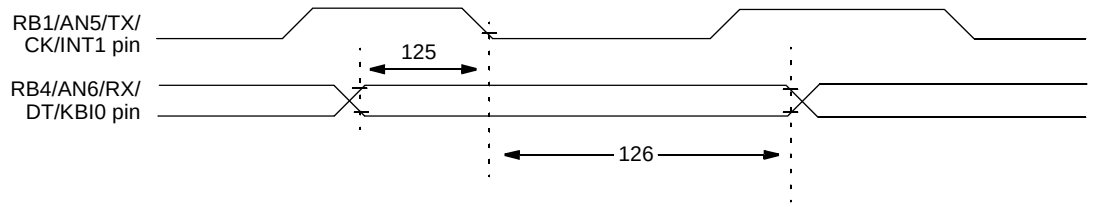


Note: Refer to Figure 22-5 for load conditions.

TABLE 22-11: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	PIC18F1X20	—	40	ns	
			PIC18LF1X20	—	100	ns	
121	Tckrf	Clock Out Rise Time and Fall Time (Master mode)	PIC18F1X20	—	20	ns	
			PIC18LF1X20	—	50	ns	
122	Tdtrf	Data Out Rise Time and Fall Time	PIC18F1X20	—	20	ns	
			PIC18LF1X20	—	50	ns	

FIGURE 22-13: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 22-5 for load conditions.

FIGURE 23-27: V_{OL} vs. I_{OL} OVER TEMPERATURE (-40°C TO +125°C), V_{DD} = 5.0V

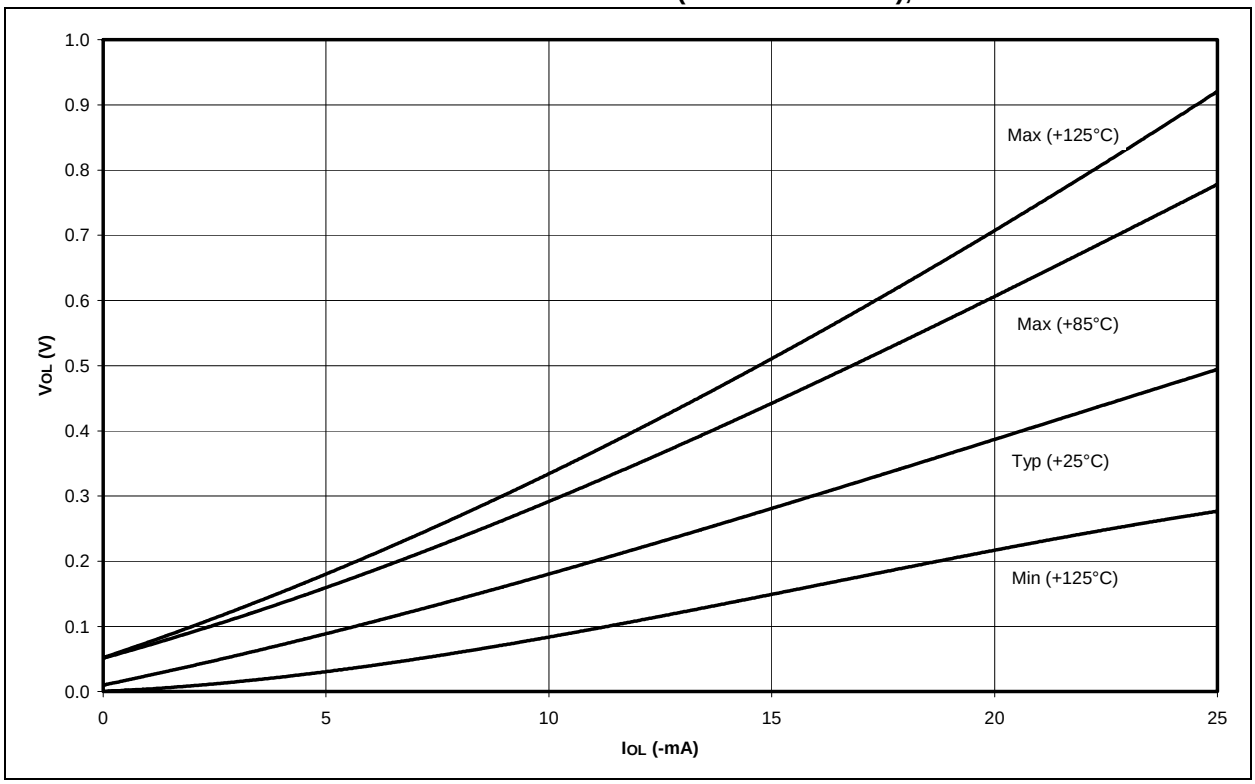


FIGURE 23-28: ΔIPD TIMER1 OSCILLATOR, -10°C TO +70°C SLEEP MODE, TMR1 COUNTER DISABLED