**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, IBT LIN/J2602 |
| Number of I/O | 12 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 9x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 20-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1829lin-i-ss |

### Peripheral Highlights (Continued)

- Data Signal Modulator Module:
  - Selectable modulator and carrier sources
- SR Latch:
  - Multiple Set/Reset input options
  - Emulates 555 Timer applications

### PIC12(L)F1822/1840/PIC16(L)F182x/1847 Family Types

| Device | Data Sheet Index | Program Memory Flash (words) | Data EEPROM (bytes) | Data SRAM (bytes) | I/O's(2) | 10-bit ADC (ch) | CapSense (ch) | Comparators | Timers (8/16-bit) | EUSART | MSSP ($I^2C$™/SPI) | ECCP (Full-Bridge) ECCP (Half-Bridge) CCP | SR Latch | Debug(1) | XLP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC12(L)F1822 | (1) | 2K | 256 | 128 | 6 | 4 | 4 | 1 | 2/1 | 1 | 1 | 0/1/0 | Y | I/H | Y |
| PIC12(L)F1840 | (2) | 4K | 256 | 256 | 6 | 4 | 4 | 1 | 2/1 | 1 | 1 | 0/1/0 | Y | I/H | Y |
| PIC16(L)F1823 | (1) | 2K | 256 | 128 | 12 | 8 | 8 | 2 | 2/1 | 1 | 1 | 1/0/0 | Y | I/H | Y |
| PIC16(L)F1824 | (3) | 4K | 256 | 256 | 12 | 8 | 8 | 2 | 4/1 | 1 | 1 | 1/1/2 | Y | I/H | Y |
| PIC16(L)F1825 | (4) | 8K | 256 | 1024 | 12 | 8 | 8 | 2 | 4/1 | 1 | 1 | 1/1/2 | Y | I/H | Y |
| PIC16(L)F1826 | (5) | 2K | 256 | 256 | 16 | 12 | 12 | 2 | 2/1 | 1 | 1 | 1/0/0 | Y | I/H | Y |
| PIC16(L)F1827 | (5) | 4K | 256 | 384 | 16 | 12 | 12 | 2 | 4/1 | 1 | 2 | 1/1/2 | Y | I/H | Y |
| PIC16(L)F1828 | (3) | 4K | 256 | 256 | 18 | 12 | 12 | 2 | 4/1 | 1 | 1 | 1/1/2 | Y | I/H | Y |
| PIC16(L)F1829 | (4) | 8K | 256 | 1024 | 18 | 12 | 12 | 2 | 4/1 | 1 | 2 | 1/1/2 | Y | I/H | Y |
| PIC16(L)F1847 | (6) | 8K | 256 | 1024 | 16 | 12 | 12 | 2 | 4/1 | 1 | 2 | 1/1/2 | Y | I/H | Y |

**Note 1:** I - Debugging, Integrated on Chip; H - Debugging, available using Debug Header.

**2:** One pin is input-only.

**Data Sheet Index:** (Unshaded devices are described in this document.)

**1:** DS41413 PIC12(L)F1822/PIC16(L)F1823 Data Sheet, 8/14-Pin Flash Microcontrollers.

**2:** DS41441 PIC12(L)F1840 Data Sheet, 8-Pin Flash Microcontrollers.

**3:** DS41419 PIC16(L)F1824/1828 Data Sheet, 28/40/44-Pin Flash Microcontrollers.

**4:** DS41440 PIC16(L)F1825/1829 Data Sheet, 14/20-Pin Flash Microcontrollers.

**5:** DS41391 PIC16(L)F1826/1827 Data Sheet, 18/20/28-Pin Flash Microcontrollers.

**6:** DS41453 PIC16(L)F1847 Data Sheet, 18/20/28-Pin Flash Microcontrollers.

**Note:** For other small form-factor package availability and marking information, please visit www.microchip.com/packaging or contact your local sales office.


© 2010-2015 Microchip Technology Inc.

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

**http://www.microchip.com**

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.


© 2010-2015 Microchip Technology Inc.

#### 3.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The High directive will set bit<7> if a label points to a location in program memory.

**EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR**

```
constants
    RETLW DATA0        ;Index0 data
    RETLW DATA1        ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;… LOTS OF CODE…
    MOVLW   LOW constants
    MOVWF   FSR1L
    MOVLW   HIGH constants
    MOVWF   FSR1H
    MOVIW   0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

## 3.2 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-2):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.5 "Indirect Addressing"** for more information.

Data Memory uses a 12-bit address. The upper seven bits of the address define the Bank address and the lower five bits select the registers/RAM in that bank.

### 3.2.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation of the PIC16(L)F1825/9. These registers are listed below:

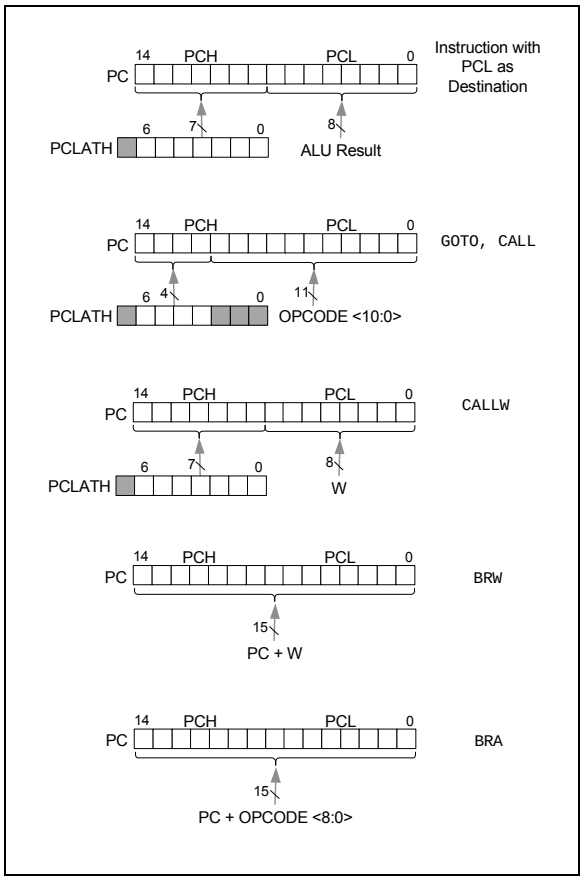
- INDF0
- INDF1
- PCL
- STATUS
- FSR0 Low
- FSR0 High
- FSR1 Low
- FSR1 High
- BSR
- WREG
- PCLATH
- INTCON

| | |
|---|---|
| **Note:** | The core registers are the first 12 addresses of every data memory bank. |

## 3.3 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 3-3 shows the five situations for the loading of the PC.

**FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS**




### 3.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

### 3.3.2 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note AN556, *"Implementing a Table Read"* (DS00556).

### 3.3.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed calls by combining PCLATH and W to form the destination address. A computed CALLW is accomplished by loading the W register with the desired address and executing CALLW. The PCL register is loaded with the value of W and PCH is loaded with PCLATH.

### 3.3.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the W register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address PC + 1 + W.

If using BRA, the entire PC will be loaded with PC + 1 +, the signed value of the operand of the BRA instruction.


© 2010-2015 Microchip Technology Inc.

## 3.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figures 3-4 through 3-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer if the STVREN bit is programmed to '0' (Configuration Word 2). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

| **Note 1:** | There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address. |
|---|---|

### 3.4.1 ACCESSING THE STACK

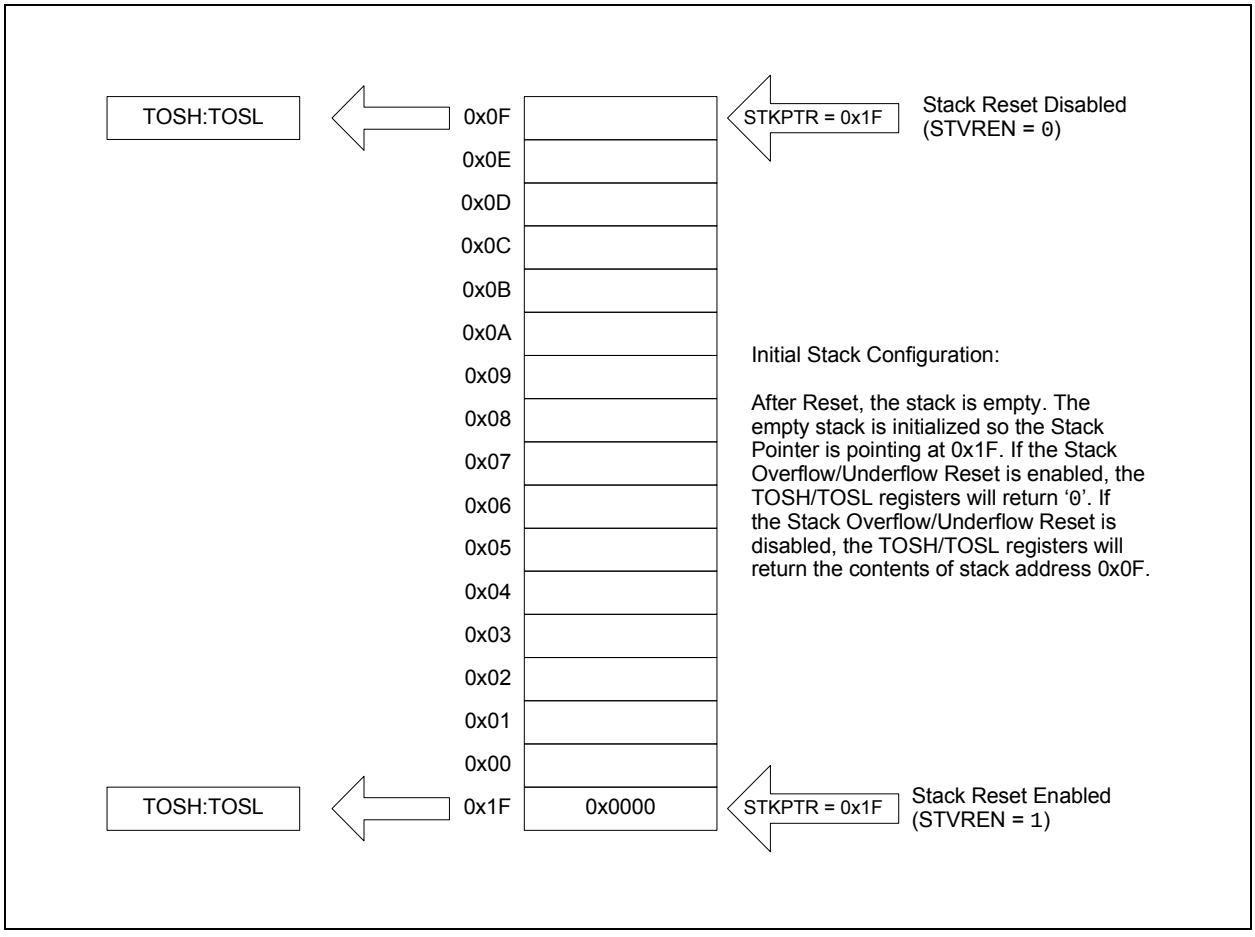
The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

| **Note:** | Care should be taken when modifying the STKPTR while interrupts are enabled. |
|---|---|

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC and then decrement the STKPTR.

Reference Figure 3-4 through Figure 3-7 for examples of accessing the stack.

**FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1**

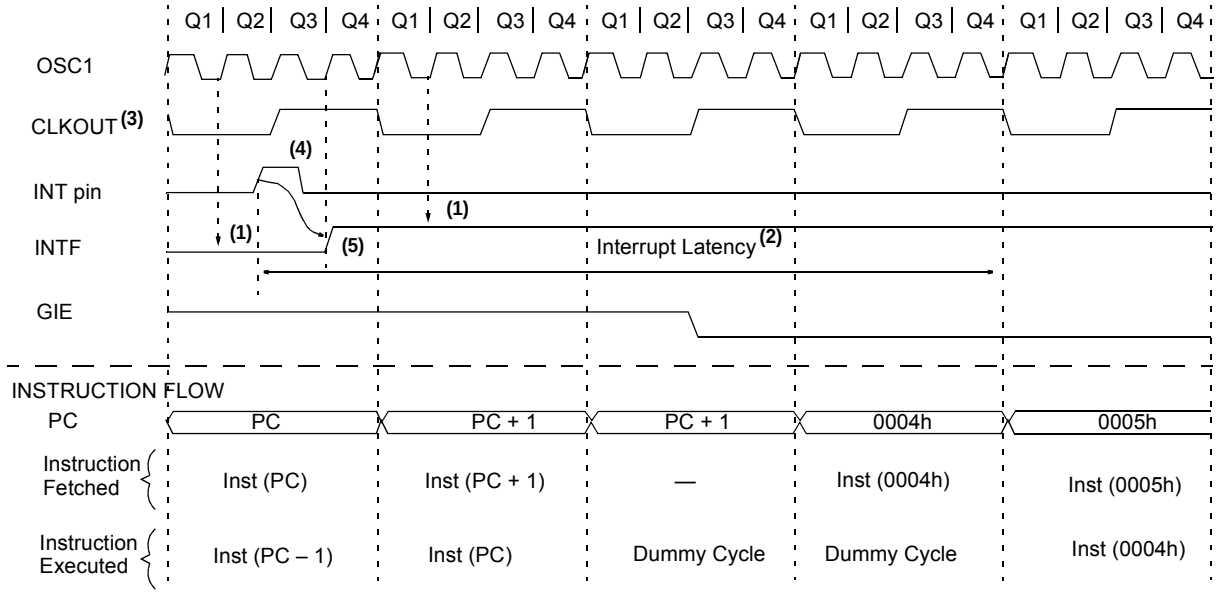


**REGISTER 4-1: CONFIGURATION WORD 1 (CONTINUED)**

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
- 111 = ECH: External Clock, High-Power mode (4-32 MHz): device clock supplied to CLKIN pin
- 110 = ECM: External Clock, Medium-Power mode (0.5-4 MHz): device clock supplied to CLKIN pin
- 101 = ECL: External Clock, Low-Power mode (0-0.5 MHz): device clock supplied to CLKIN pin
- 100 = INTOSC oscillator: I/O function on CLKIN pin
- 011 = EXTRC oscillator: External RC circuit connected to CLKIN pin
- 010 = HS oscillator: High-speed crystal/resonator connected between OSC1 and OSC2 pins
- 001 = XT oscillator: Crystal/resonator connected between OSC1 and OSC2 pins
- 000 = LP oscillator: Low-power crystal connected between OSC1 and OSC2 pins

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.
**2:** The entire data EEPROM will be erased when the code protection is turned off during an erase.
**3:** The entire program memory will be erased when the code protection is turned off.

**FIGURE 8-3: INT PIN INTERRUPT TIMING**




**Note 1:** INTF flag is sampled here (every Q1).

**2:** Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

**3:** CLKOUT not available in all oscillator modes.

**4:** For minimum width of INT pulse, refer to AC specifications in **Section 30.0 "Electrical Specifications"**.

**5:** INTF is enabled to be set any time during the Q4-Q1 cycles.

**EXAMPLE 11-5: WRITING TO FLASH PROGRAM MEMORY**

```
; This write routine assumes the following:
; 1. The 16 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F
;
        BCF         INTCON,GIE     ; Disable ints so required sequences will execute properly
        BANKSEL     EEADRH         ; Bank 3
        MOVF        ADDRH,W        ; Load initial address
        MOVWF       EEADRH         ;
        MOVF        ADDRL,W        ;
        MOVWF       EEADRL         ;
        MOVLW       LOW DATA_ADDR  ; Load initial data address
        MOVWF       FSR0L          ;
        MOVLW       HIGH DATA_ADDR ; Load initial data address
        MOVWF       FSR0H          ;
        BSF         EECON1,EEPGD   ; Point to program memory
        BCF         EECON1,CFGS    ; Not configuration space
        BSF         EECON1,WREN    ; Enable writes
        BSF         EECON1,LWLO    ; Only Load Write Latches
LOOP
        MOVIW       FSR0++         ; Load first data byte into lower
        MOVWF       EEDATL         ;
        MOVIW       FSR0++         ; Load second data byte into upper
        MOVWF       EEDATH         ;

        MOVF        EEADRL,W       ; Check if lower bits of address are '000'
        XORLW       0x07           ; Check if we're on the last of 8 addresses
        ANDLW       0x07           ;
        BTFSC       STATUS,Z       ; Exit if last of eight words,
        GOTO        START_WRITE    ;

  Required Sequence
        MOVLW       55h            ; Start of required write sequence:
        MOVWF       EECON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       EECON2         ; Write AAh
        BSF         EECON1,WR      ; Set WR bit to begin write
        NOP                        ; Any instructions here are ignored as processor
                                   ; halts to begin write sequence
        NOP                        ; Processor will stop here and wait for write to complete.

                                   ; After write processor continues with 3rd instruction.

        INCF        EEADRL,F       ; Still loading latches Increment address
        GOTO        LOOP           ; Write next latches

START_WRITE
        BCF         EECON1,LWLO    ; No more loading latches - Actually start Flash program
                                   ; memory write

  Required Sequence
        MOVLW       55h            ; Start of required write sequence:
        MOVWF       EECON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       EECON2         ; Write AAh
        BSF         EECON1,WR      ; Set WR bit to begin write
        NOP                        ; Any instructions here are ignored as processor
                                   ; halts to begin write sequence
        NOP                        ; Processor will stop here and wait for write complete.

                                   ; after write processor continues with 3rd instruction
        BCF         EECON1,WREN    ; Disable writes
        BSF         INTCON,GIE     ; Enable interrupts
```


© 2010-2015 Microchip Technology Inc.

**FIGURE 23-4: CARRIER LOW SYNCHRONIZATION (MDSHSYNC = 0, MDCLSYNC = 1)**

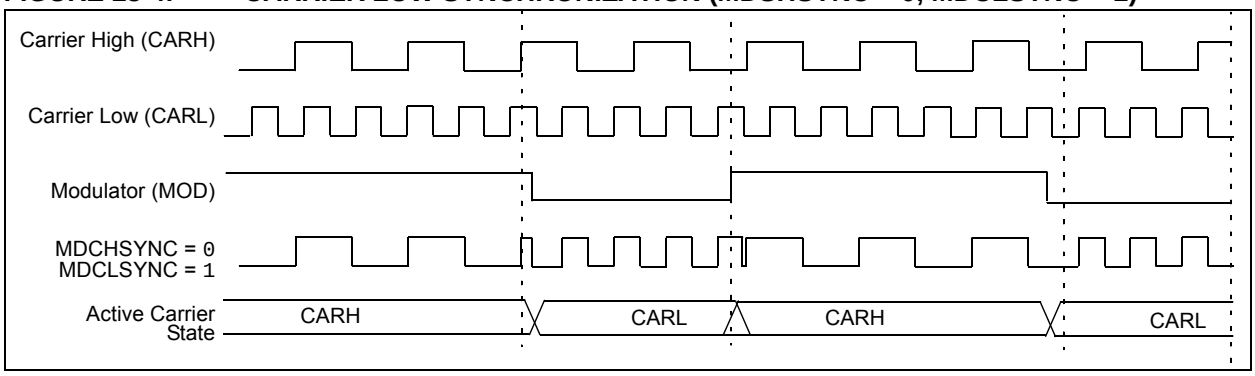



**FIGURE 23-5: FULL SYNCHRONIZATION (MDSHSYNC = 1, MDCLSYNC = 1)**

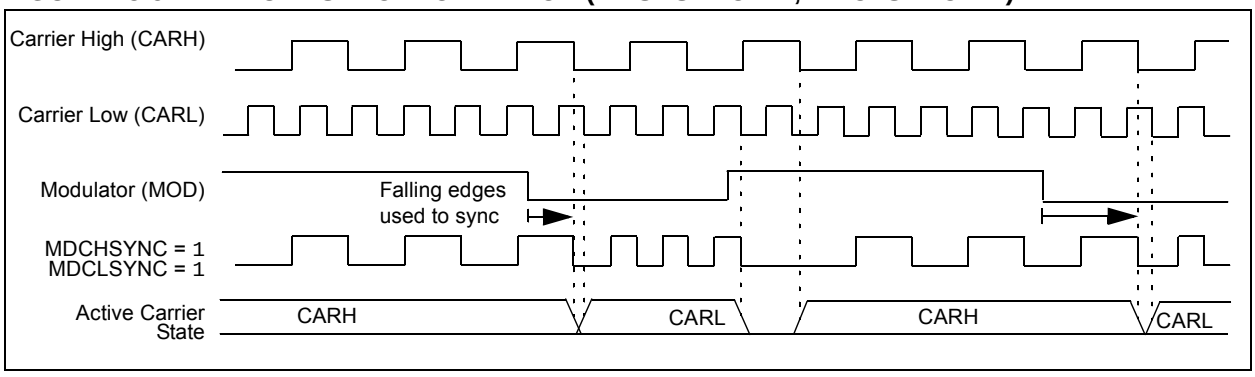
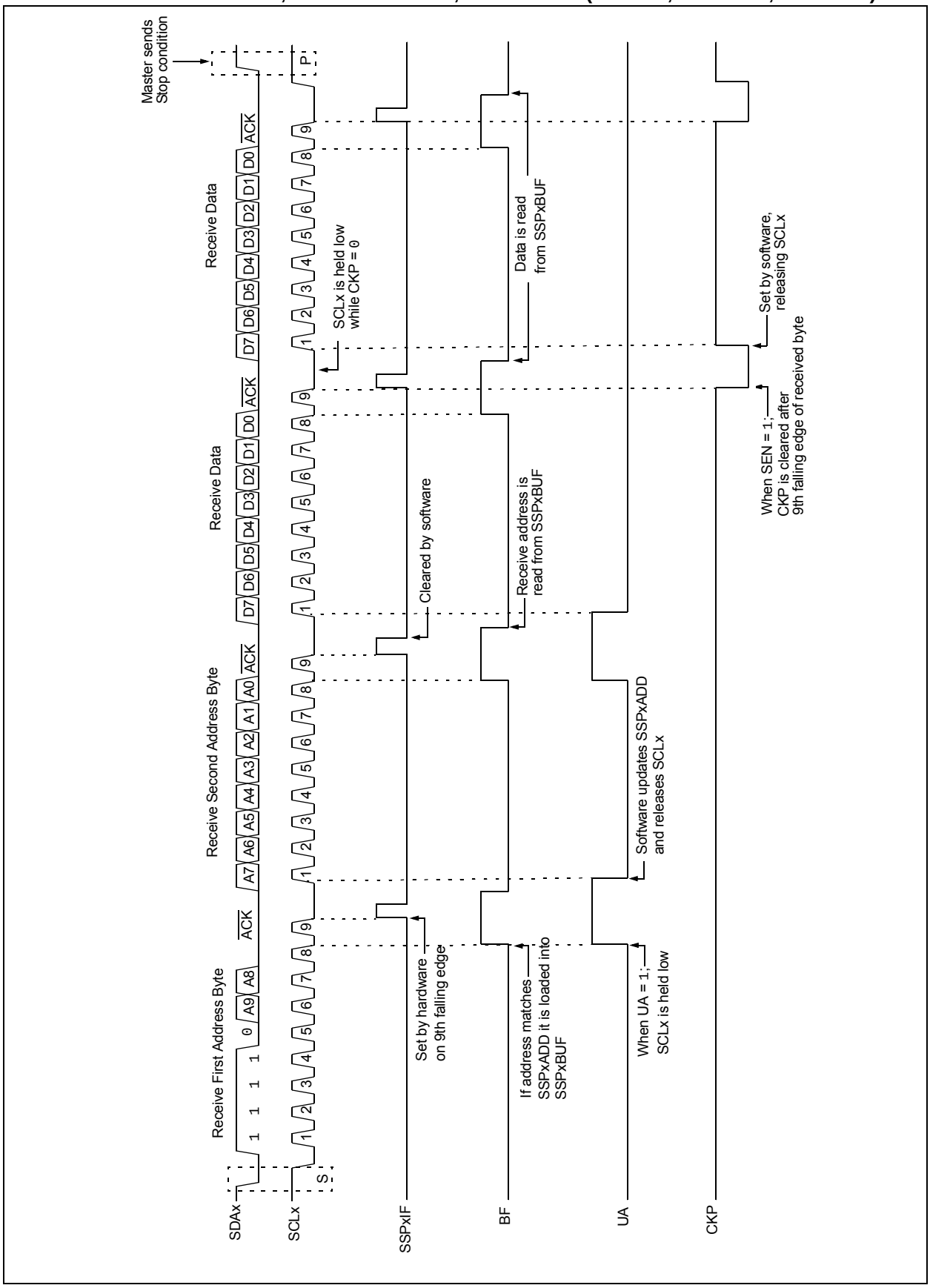


**FIGURE 25-20: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**



### 25.5.7 GENERAL CALL ADDRESS SUPPORT

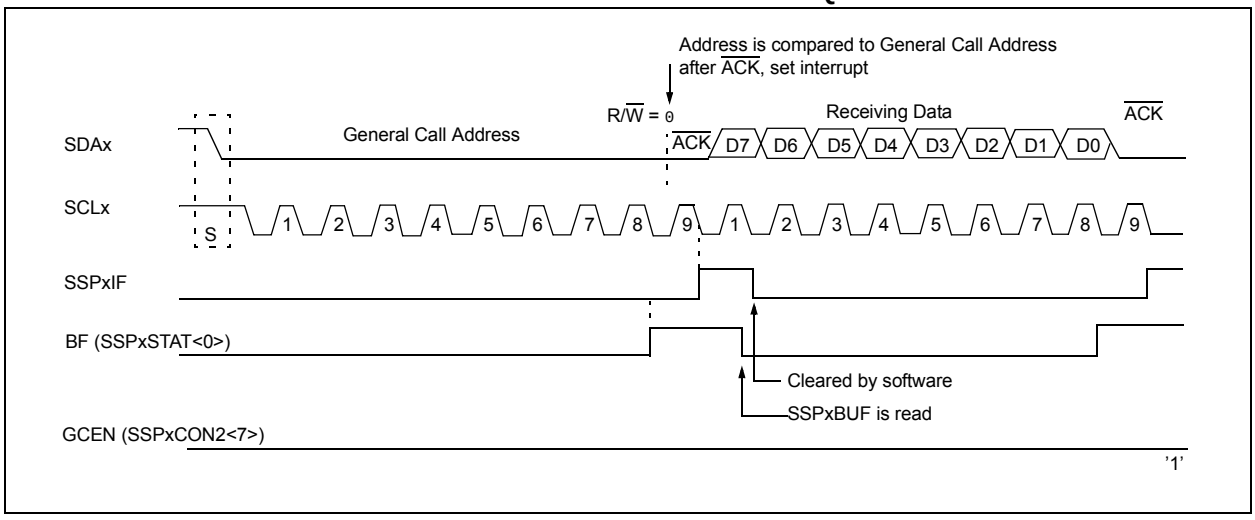
The addressing procedure for the $I^2C$ bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the $I^2C$ protocol, defined as address 0x00. When the GCEN bit of the SSPxCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/$\overline{W}$ bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. Figure 25-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCLx. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

**FIGURE 25-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE**




### 25.5.8 SSPX MASK REGISTER

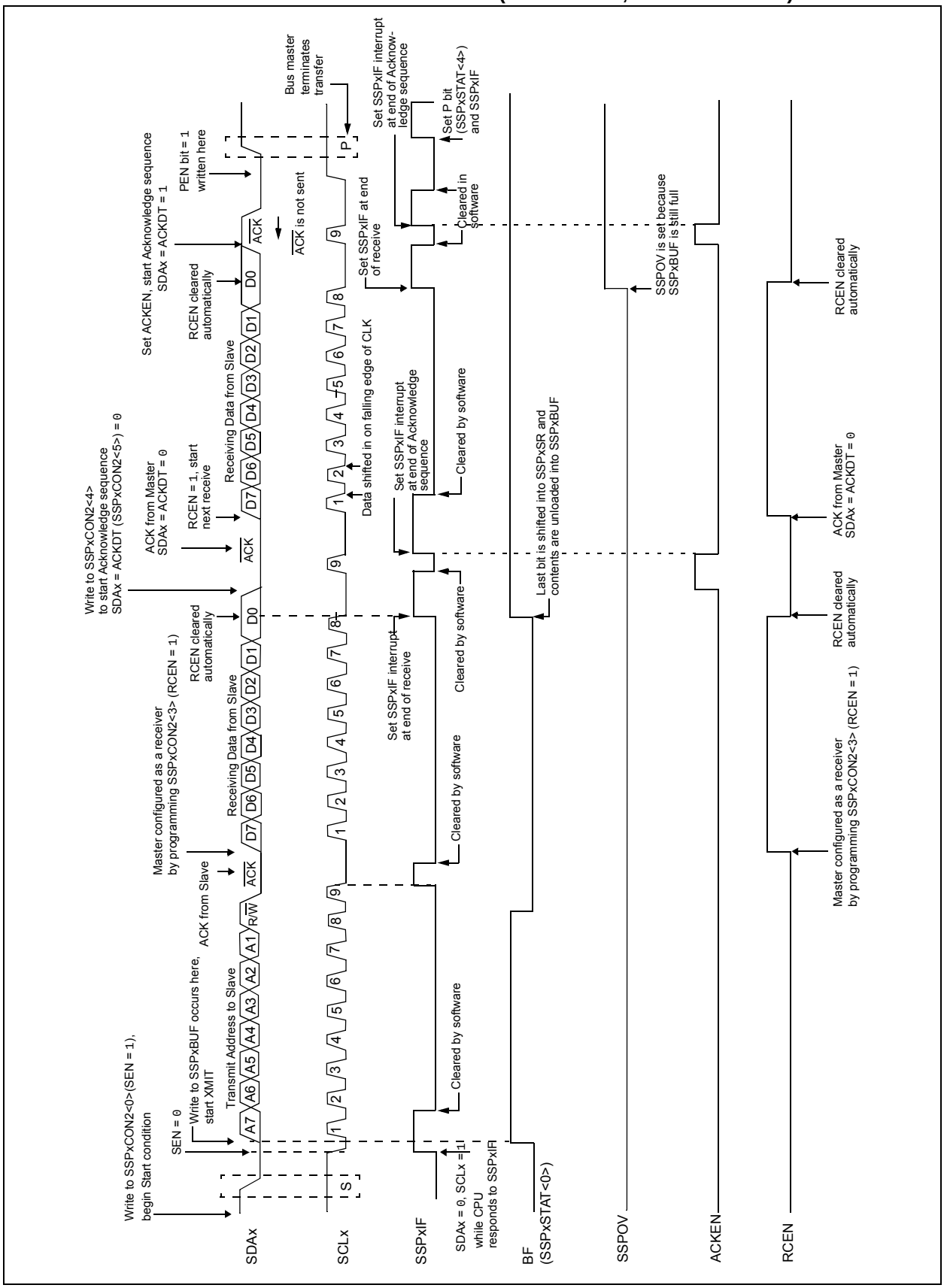
An SSPx Mask (SSPMSK) register (Register 25-5) is available in $I^2C$ Slave mode as a mask for the value held in the SSPxSR register during an address comparison operation. A zero (‘0’) bit in the SSPMSK register has the effect of making the corresponding bit of the received address a “don’t care”.

This register is reset to all ‘1’s upon any Reset condition and, therefore, has no effect on standard SSPx operation until written with a mask value.

The SSPx Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSPx mask has no effect during the reception of the first (high) byte of the address.

**FIGURE 25-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)**



#### 26.1.2.8 Asynchronous Reception Setup:

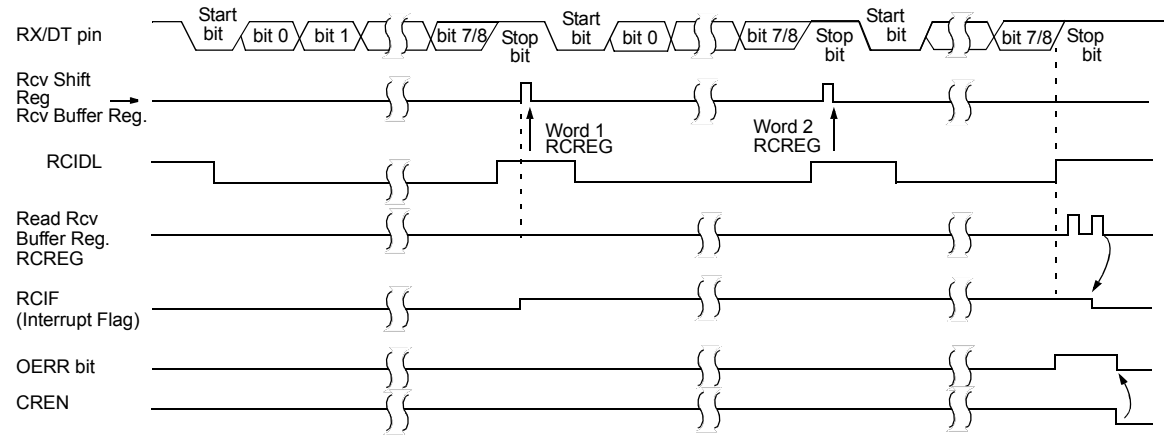
1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 26.3 "EUSART Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RCIE bit of the PIE1 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 RCIF 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 RCIE interrupt enable bit was also set.
8. Read the RCSTA 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 RCREG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

#### 26.1.2.9 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 SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 26.3 "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 RCIE bit of the PIE1 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 RCIF 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 RCIE interrupt enable bit was also set.
9. Read the RCSTA 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 RCREG 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.

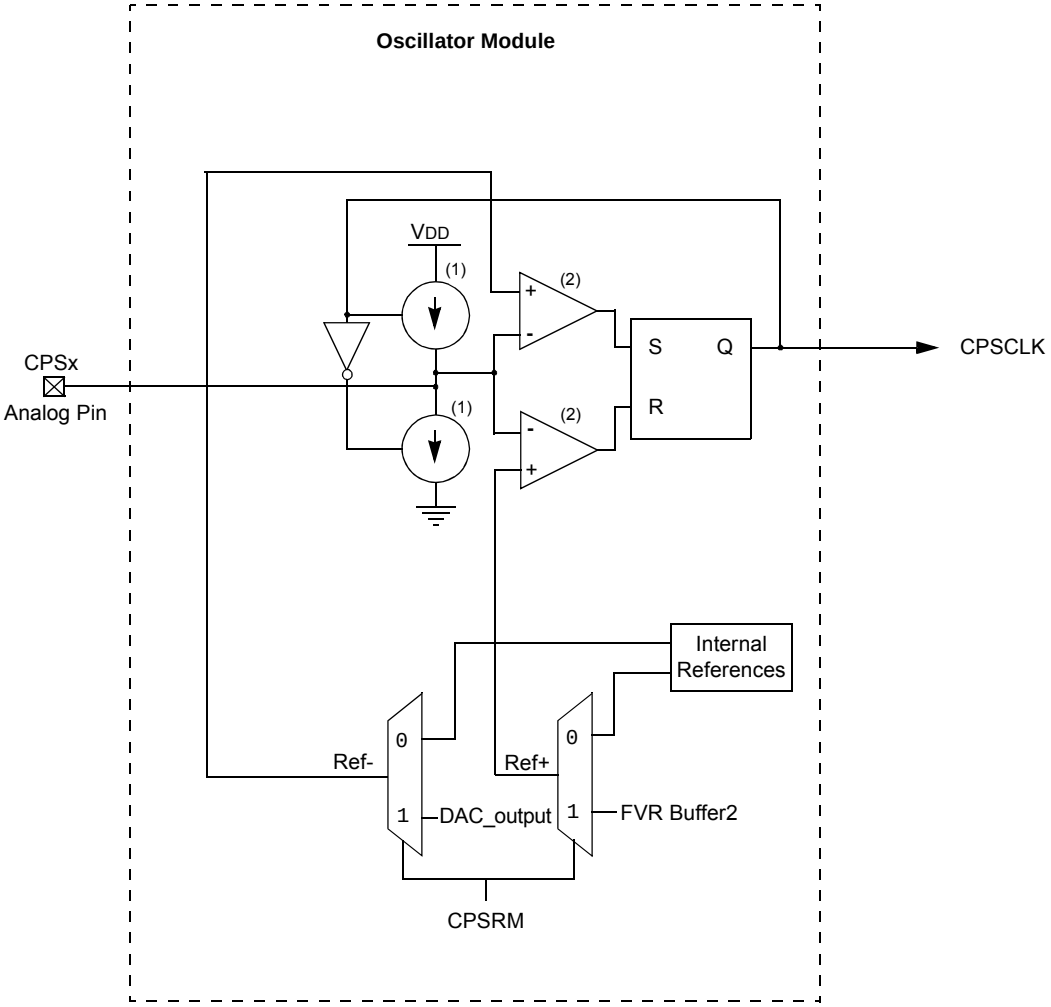
**FIGURE 26-5: ASYNCHRONOUS RECEPTION**




**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.


© 2010-2015 Microchip Technology Inc.

**FIGURE 27-2: CAPACITIVE SENSING OSCILLATOR BLOCK DIAGRAM**




**Note 1:** Module Enable and Current mode selections are not shown.
**2:** Comparators remain active in Noise Detection mode.


© 2010-2015 Microchip Technology Inc.

**RRF** **Rotate Right f through Carry**

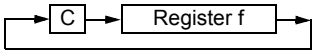
Syntax: [ *label* ] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.




**SLEEP** **Enter Sleep mode**

Syntax: [ *label* ] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

**SUBLW** **Subtract W from literal**

Syntax: [ *label* ] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

| | |
|---|---|
| C = 0 | W > k |
| C = 1 | W ≤ k |
| DC = 0 | W<3:0> > k<3:0> |
| DC = 1 | W<3:0> ≤ k<3:0> |

**SUBWF** **Subtract W from f**

Syntax: [ *label* ] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f.

| | |
|---|---|
| C = 0 | W > f |
| C = 1 | W ≤ f |
| DC = 0 | W<3:0> > f<3:0> |
| DC = 1 | W<3:0> ≤ f<3:0> |

**SUBWFB** **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – (W) – ($\overline{B}$) → dest

Status Affected: C, DC, Z

Description: Subtract W and the BORROW flag (CARRY) 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'.


© 2010-2015 Microchip Technology Inc.

## 30.4 DC Characteristics: PIC16(L)F1825/9-I/E

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O PORT: | | | | | |
| D030 | | with TTL buffer | — | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | — | — | 0.15 VDD | V | 1.8V ≤ VDD ≤ 4.5V |
| D031 | | with Schmitt Trigger buffer | — | — | 0.2 VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| | | with I²C™ levels | — | — | 0.3 VDD | V | |
| | | with SMBus levels | — | — | 0.8 | V | 2.7V ≤ VDD ≤ 5.5V |
| D032 | | $\overline{MCLR}$, OSC1 (RC mode)(1) | — | — | 0.2 VDD | V | |
| D033 | | OSC1 (HS mode) | — | — | 0.3 VDD | V | |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O ports: | | | | | |
| D040 | | with TTL buffer | 2.0 | — | — | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8 | — | — | V | 1.8V ≤ VDD ≤ 4.5V |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | — | V | 2.0V ≤ VDD ≤ 5.5V |
| | | with I²C™ levels | 0.7 VDD | — | — | V | |
| | | with SMBus levels | 2.1 | — | — | V | 2.7V ≤ VDD ≤ 5.5V |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | — | V | |
| D043A | | OSC1 (HS mode) | 0.7 VDD | — | — | V | |
| D043B | | OSC1 (RC mode) | 0.9 VDD | — | — | V | VDD > 2.0V **(Note 1)** |
| | IIL | **Input Leakage Current(2)** | | | | | |
| D060 | | I/O ports | — | ± 5 | ± 125 | nA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance at 85°C |
| | | | | ± 5 | ± 1000 | nA | 125°C |
| D061 | | $\overline{MCLR}$(3) | — | ± 50 | ± 200 | nA | VSS ≤ VPIN ≤ VDD at 85°C |
| | IPUR | **Weak Pull-up Current** | | | | | |
| D070* | | | 25<br>25 | 100<br>140 | 200<br>300 | μA | VDD = 3.3V, VPIN = VSS<br>VDD = 5.0V, VPIN = VSS |
| | VOL | **Output Low Voltage(4)** | | | | | |
| D080 | | I/O ports | — | — | 0.6 | V | IOL = 8mA, VDD = 5V<br>IOL = 6mA, VDD = 3.3V<br>IOL = 1.8mA, VDD = 1.8V |
| | VOH | **Output High Voltage(4)** | | | | | |
| D090 | | I/O ports | VDD - 0.7 | — | — | V | IOH = 3.5mA, VDD = 5V<br>IOH = 3mA, VDD = 3.3V<br>IOH = 1mA, VDD = 1.8V |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D101* | COSC2 | OSC2 pin | — | — | 15 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101A* | CIO | All I/O pins | — | — | 50 | pF | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** Including OSC2 in CLKOUT mode.


© 2010-2015 Microchip Technology Inc.

## 30.5 Memory Programming Requirements

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
| | | **Program Memory High Voltage Programming Specifications** | | | | | |
| D110 | VIHH | Voltage on MCLR/VPP/RA5 pin | 8.0 | — | 9.0 | V | **(Note 3, 4)** |
| D111 | IDDVPP | Programming/Erase Current on VPP, High Voltage Programming | — | — | 10 | mA | |
| D112 | VBE | VDD for Bulk Erase | 2.7 | — | VDDMAX | V | |
| D113 | VPEW | VDD for Write or Row Erase | VDDMIN | — | VDDMAX | V | |
| D114 | IPPPGM | Programming/Erase Current on VPP, Low Voltage Programming | — | 1.0 | — | mA | |
| D115 | IDDPGM | Programming/Erase Current on VDD, High or Low Voltage Programming | — | 5.0 | — | mA | |
| | | **Data EEPROM Memory** | | | | | |
| D116 | ED | Byte Endurance | 100K | — | — | E/W | -40°C to +85°C |
| D117 | VDRW | VDD for Read/Write | VDDMIN | — | VDDMAX | V | |
| D118 | TDEW | Erase/Write Cycle Time | — | 4.0 | 5.0 | ms | |
| D119 | TRETD | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |
| D120 | TREF | Number of Total Erase/Write Cycles before Refresh(2) | 1M | 10M | — | E/W | -40°C to +85°C |
| | | **Program Flash Memory** | | | | | |
| D121 | EP | Cell Endurance | 10K | — | — | E/W | -40°C to +85°C (**Note 1**) |
| D122 | VPRW | VDD for Read/Write | VDDMIN | — | VDDMAX | V | |
| D123 | TIW | Self-timed Write Cycle Time | — | 2 | 2.5 | ms | |
| D124 | TRETD | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Self-write and Block Erase.

**2:** Refer to **Section 11.2 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

**3:** Required only if single-supply programming is disabled.

**4:** The MPLAB ICD 2 does not support variable VPP output. Circuitry to limit the MPLAB ICD 2 VPP voltage must be placed between the MPLAB ICD 2 and target system when programming or debugging with the MPLAB ICD 2.

**TABLE 30-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2 | — | — | μs | |
| 31 | TWDTLP | Low-Power Watchdog Timer Time-out Period | 12 | 16 | 20 | ms | VDD = 3.3V-5V, 1:16 Prescaler used |
| 32 | TOST | Oscillator Start-up Timer Period(1) | — | 1024 | — | Tosc | |
| 33* | TPWRT | Power-up Timer Period, $\overline{\text{PWRTE}}$ = 0 | 40 | 65 | 140 | ms | |
| 34* | TIOZ | I/O high-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | μs | |
| 35 | VBOR | Brown-out Reset Voltage(2) | 2.55<br>1.80 | 2.70<br>1.9 | 2.85<br>2.05 | V<br>V | BORV = 0<br>BORV = 1 |
| 36* | VHYST | Brown-out Reset Hysteresis | 20 | 35 | 75 | mV | -40°C to +85°C |
| 37* | TBORDC | Brown-out Reset DC Response Time | 1 | 3 | 35 | μs | VDD ≤ VBOR |

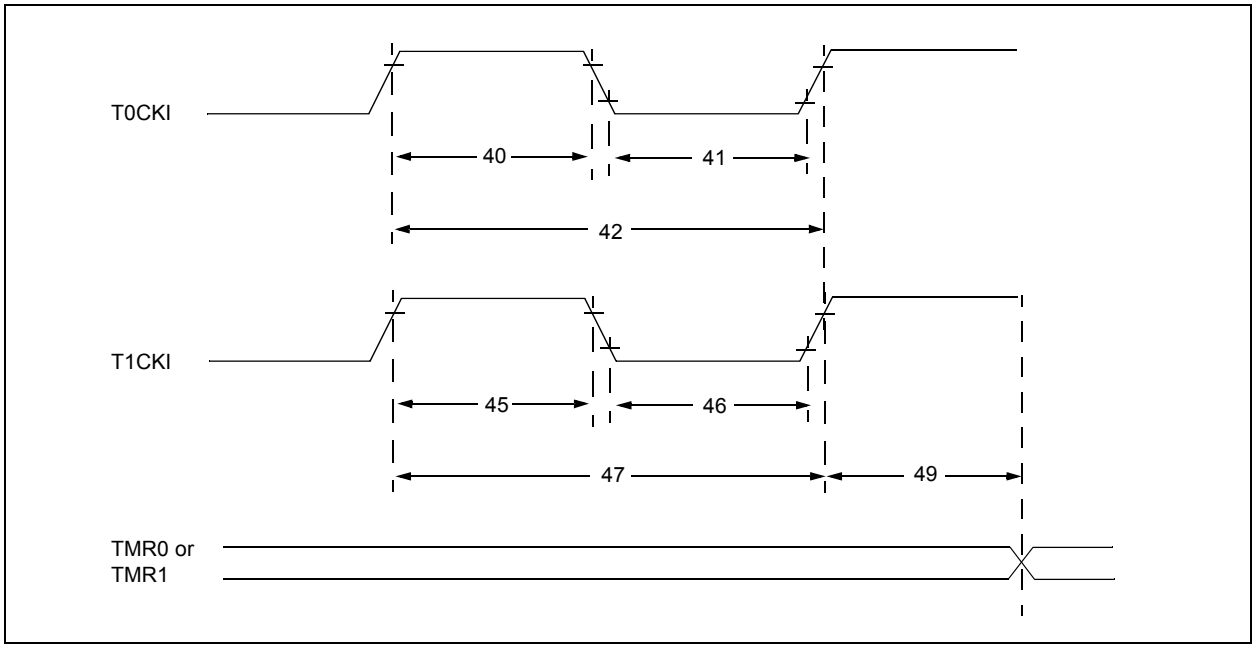
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** By design, the Oscillator Start-up (OST) counts the first 1024 cycles, independent of frequency.

**2:** To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

**FIGURE 30-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

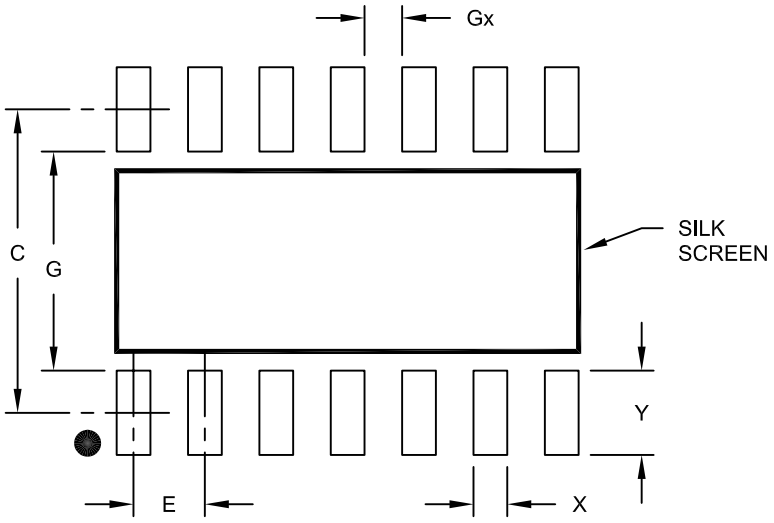




© 2010-2015 Microchip Technology Inc.

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 5.40 | |
| Contact Pad Width | X | | | 0.60 |
| Contact Pad Length | Y | | | 1.50 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 3.90 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

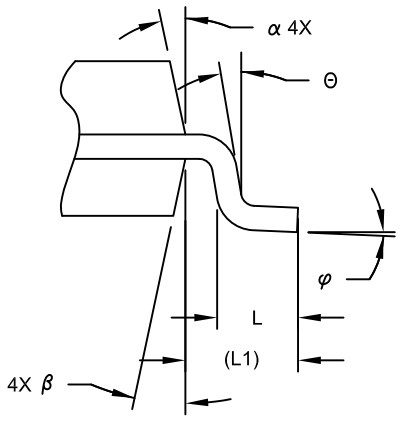
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

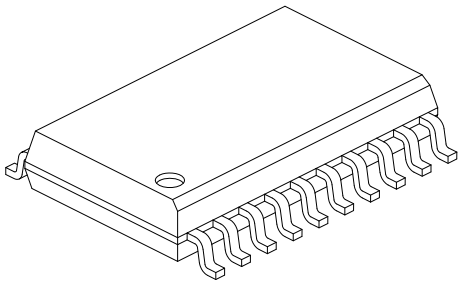

© 2010-2015 Microchip Technology Inc.

## 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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




**VIEW C**



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 20 | |
| Pitch | e | | 1.27 BSC | |
| Overall Height | A | - | - | 2.65 |
| Molded Package Thickness | A2 | 2.05 | - | - |
| Standoff § | A1 | 0.10 | - | 0.30 |
| Overall Width | E | | 10.30 BSC | |
| Molded Package Width | E1 | | 7.50 BSC | |
| Overall Length | D | | 12.80 BSC | |
| Chamfer (Optional) | h | 0.25 | - | 0.75 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | | 1.40 REF | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.20 | - | 0.33 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

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

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2