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	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf442t-i-ml

PIC18FXX2

FIGURE 4-7: DATA MEMORY MAP FOR PIC18F252/452
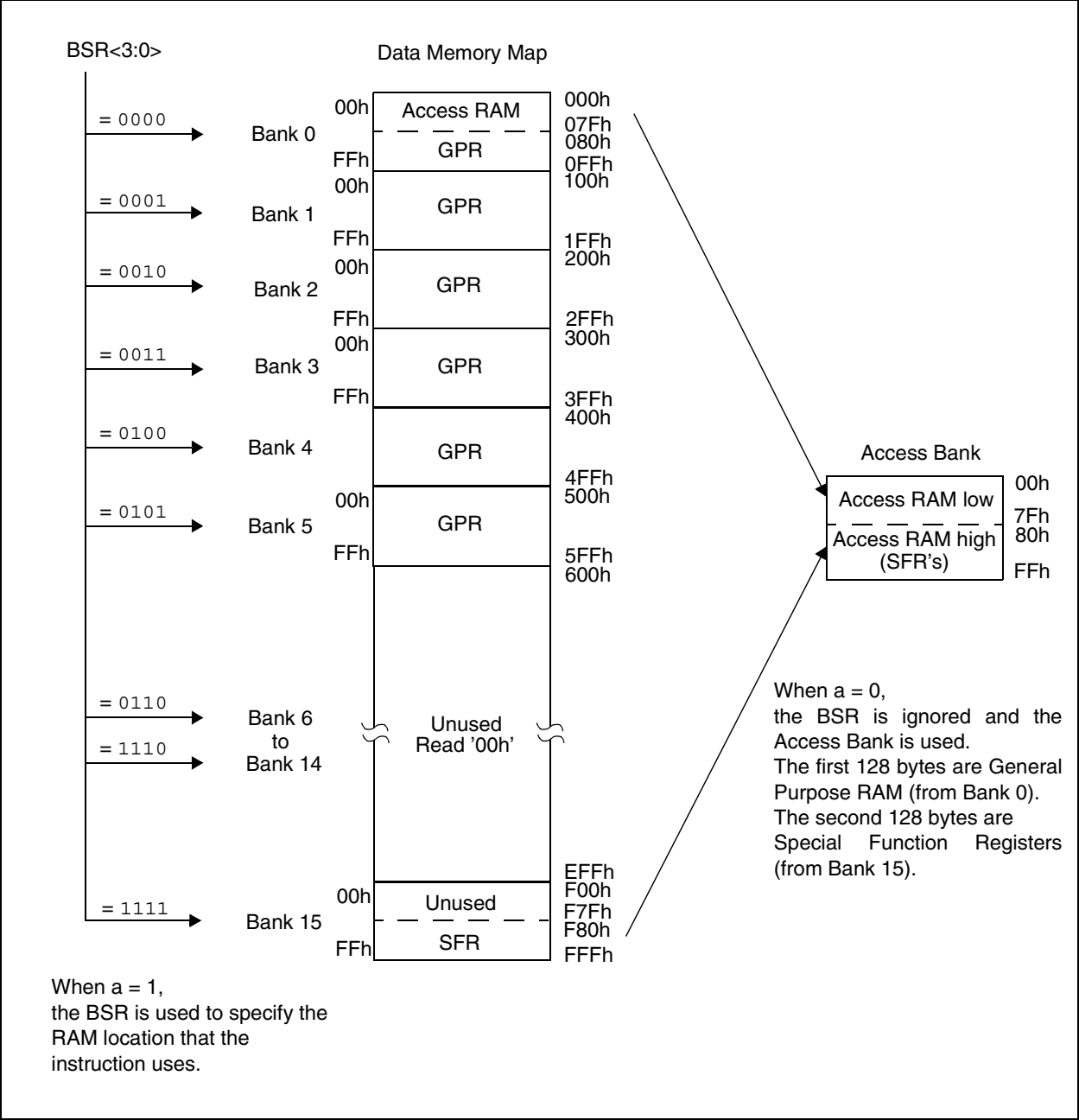
EXAMPLE 5-3: **WRITING TO FLASH PROGRAM MEMORY (CONTINUED)**

```
PROGRAM_MEMORY
          BSF       EECON1,EEPGD       ; point to FLASH program memory
          BCF       EECON1,CFGS        ; access FLASH program memory
          BSF       EECON1,WREN        ; enable write to memory
          BCF       INTCON,GIE         ; disable interrupts
          MOVLW     55h
Required  MOVWF     EECON2             ; write 55h
Sequence  MOVLW     AAh
          MOVWF     EECON2             ; write AAh
          BSF       EECON1,WR          ; start program (CPU stall)
          BSF       INTCON,GIE         ; re-enable interrupts
          DECFSZ    COUNTER_HI         ; loop until done
          BRA       PROGRAM_LOOP
          BCF       EECON1,WREN        ; disable write to memory
```

5.5.2   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.

5.5.3   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. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ Reset, or a WDT Time-out Reset during normal operation. In these situations, users can check the WRERR bit and rewrite the location.

5.5.4   PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to FLASH program memory, the write initiate sequence must also be followed. See "Special Features of the CPU" (Section 19.0) for more detail.

5.6   FLASH Program Operation During Code Protection

See "Special Features of the CPU" (Section 19.0) for details on code protection of FLASH program memory.

TABLE 5-2:   REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Address	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
FF8h	TBLPTRU	—	—	bit21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					--00 0000	--00 0000
FF7h	TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								0000 0000	0000 0000
FF6h	TBLPTRL	Program Memory Table Pointer High Byte (TBLPTR<7:0>)								0000 0000	0000 0000
FF5h	TABLAT	Program Memory Table Latch								0000 0000	0000 0000
FF2h	INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
FA7h	EECON2	EEPROM Control Register2 (not a physical register)								—	—
FA6h	EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	uu-0 u000
FA2h	IPR2	—	—	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	---1 1111	---1 1111
FA1h	PIR2	—	—	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	---0 0000	---0 0000
FA0h	PIE2	—	—	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	---0 0000	---0 0000

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented read as '0'.
            Shaded cells are not used during FLASH/EEPROM access.
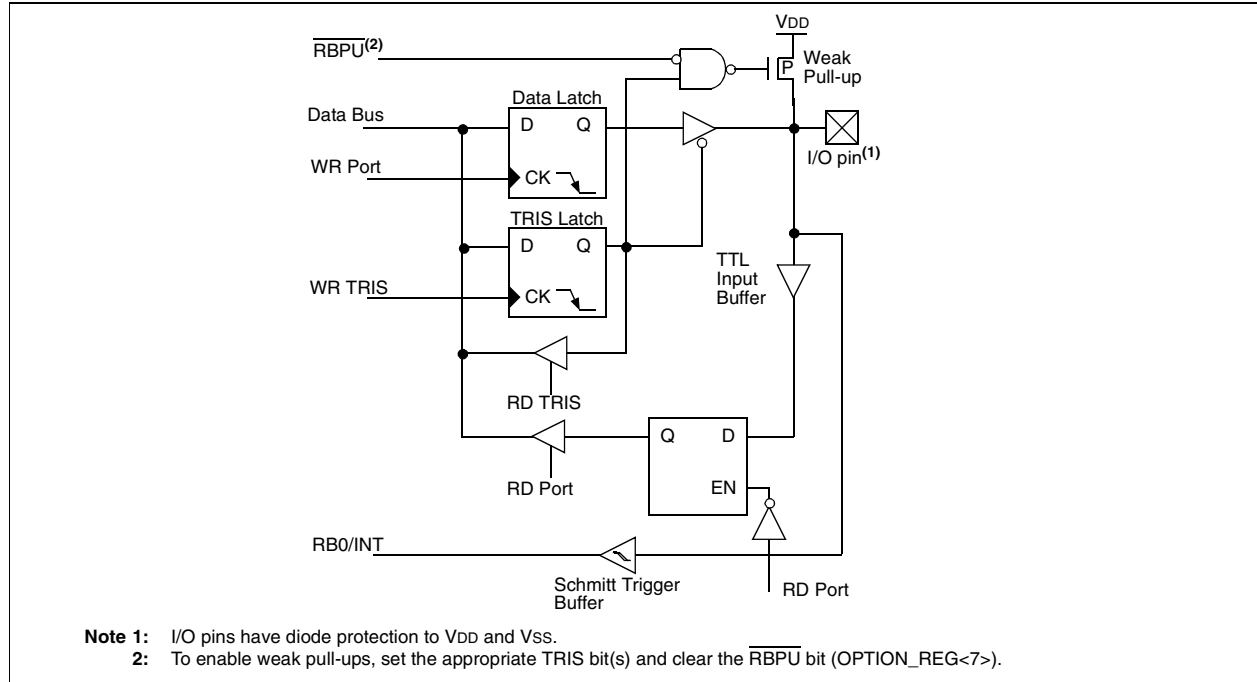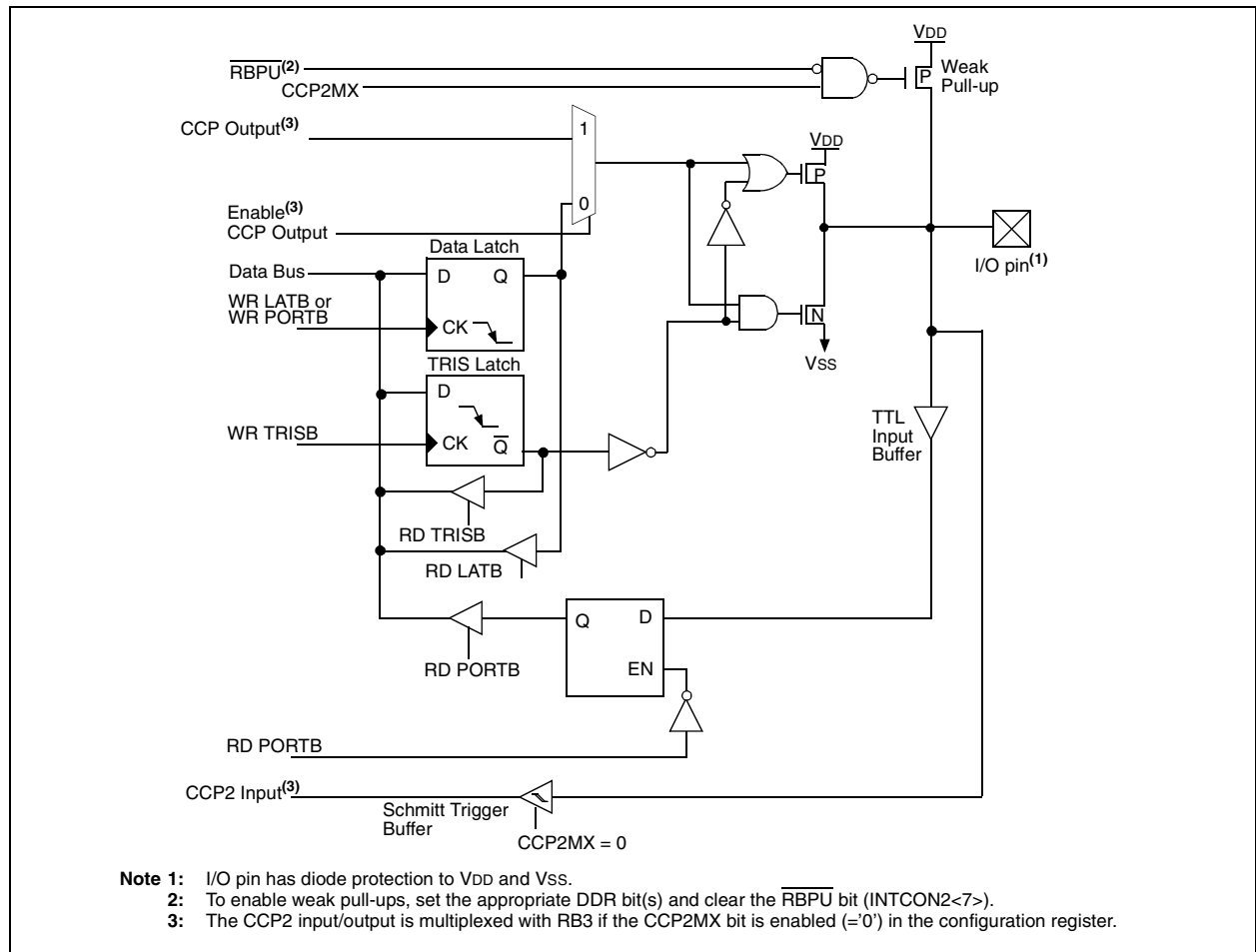
FIGURE 9-5: **BLOCK DIAGRAM OF RB2:RB0 PINS**

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 RBPU bit (OPTION_REG<7>).

FIGURE 9-6: **BLOCK DIAGRAM OF RB3 PIN**

Note 1: I/O pin has diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate DDR bit(s) and clear the RBPU bit (INTCON2<7>).
3: The CCP2 input/output is multiplexed with RB3 if the CCP2MX bit is enabled (='0') in the configuration register.

PIC18FXX2

9.6   Parallel Slave Port

The Parallel Slave Port is implemented on the 40-pin devices only (PIC18F4X2).

PORTD operates as an 8-bit wide Parallel Slave Port, or microprocessor port when control bit, PSPMODE (TRISE<4>) is set. It is asynchronously readable and writable by the external world through $\overline{RD}$ control input pin, RE0/$\overline{RD}$ and $\overline{WR}$ control input pin, RE1/$\overline{WR}$.

It can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/$\overline{RD}$ to be the $\overline{RD}$ input, RE1/$\overline{WR}$ to be the $\overline{WR}$ input and RE2/$\overline{CS}$ to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). The A/D port configuration bits PCFG2:PCFG0 (ADCON1<2:0>) must be set, which will configure pins RE2:RE0 as digital I/O.

A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low. A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are first detected low.

The PORTE I/O pins become control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs), and the ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

FIGURE 9-10:   PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)

Note: I/O pin has protection diodes to VDD and VSS.

FIGURE 9-11:   PARALLEL SLAVE PORT WRITE WAVEFORMS

© 2006 Microchip Technology Inc.

11.0   TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter
  (two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt-on-overflow from FFFFh to 0000h
- RESET from CCP module special event trigger

Figure 11-1 is a simplified block diagram of the Timer1 module.

Register 11-1 details the Timer1 control register. This register controls the Operating mode of the Timer1 module, and contains the Timer1 oscillator enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit TMR1ON (T1CON<0>).

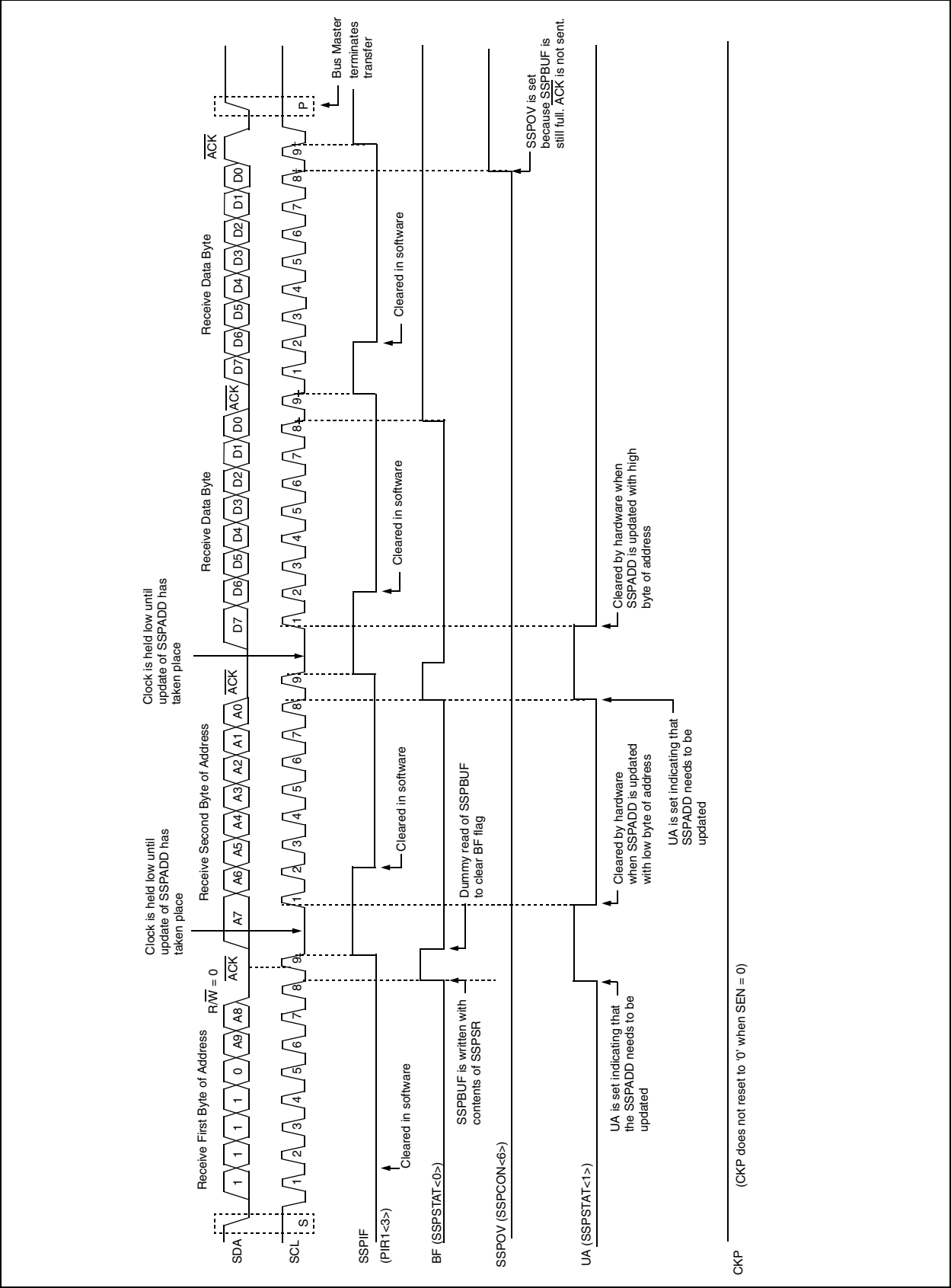### REGISTER 11-1:   T1CON: TIMER1 CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

bit 7   **RD16:** 16-bit Read/Write Mode Enable bit

   1 = Enables register Read/Write of Timer1 in one 16-bit operation
   0 = Enables register Read/Write of Timer1 in two 8-bit operations

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

bit 5-4   **T1CKPS1:T1CKPS0**: Timer1 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   **T1OSCEN:** Timer1 Oscillator Enable bit

   1 = Timer1 Oscillator is enabled
   0 = Timer1 Oscillator is shut-off
      The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2   **T1SYNC:** Timer1 External Clock Input Synchronization Select bit

   <u>When TMR1CS = 1:</u>
   1 = Do not synchronize external clock input
   0 = Synchronize external clock input

   <u>When TMR1CS = 0:</u>
   This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1   **TMR1CS:** Timer1 Clock Source Select bit

   1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
   0 = Internal clock (FOSC/4)

bit 0   **TMR1ON:** Timer1 On bit

   1 = Enables Timer1
   0 = Stops Timer1

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

PIC18FXX2

FIGURE 15-10: I²C SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

18.1 Control Register

The Low Voltage Detect Control register controls the operation of the Low Voltage Detect circuitry.

REGISTER 18-1: LVDCON REGISTER

U-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0

bit 7   bit 0

bit 7-6   **Unimplemented:** Read as '0'

bit 5   **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the Low Voltage Detect logic will generate the interrupt flag at the specified voltage range
0 = Indicates that the Low Voltage Detect logic will not generate the interrupt flag at the specified voltage range and the LVD interrupt should not be enabled

bit 4   **LVDEN:** Low Voltage Detect Power Enable bit
1 = Enables LVD, powers up LVD circuit
0 = Disables LVD, powers down LVD circuit

bit 3-0   **LVDL3:LVDL0:** Low Voltage Detection Limit bits
1111 = External analog input is used (input comes from the LVDIN pin)
1110 = 4.5V - 4.77V
1101 = 4.2V - 4.45V
1100 = 4.0V - 4.24V
1011 = 3.8V - 4.03V
1010 = 3.6V - 3.82V
1001 = 3.5V - 3.71V
1000 = 3.3V - 3.50V
0111 = 3.0V - 3.18V
0110 = 2.8V - 2.97V
0101 = 2.7V - 2.86V
0100 = 2.5V - 2.65V
0011 = 2.4V - 2.54V
0010 = 2.2V - 2.33V
0001 = 2.0V - 2.12V
0000 = Reserved

Note: LVDL3:LVDL0 modes which result in a trip point below the valid operating voltage of the device are not tested.

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

REGISTER 19-10: CONFIGURATION REGISTER 7 LOW (CONFIG7L: BYTE ADDRESS 30000Ch)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	EBTR3[1]	EBTR2[1]	EBTR1	EBTR0
bit 7							bit 0

bit 7-4   **Unimplemented:** Read as '0'

bit 3   **EBTR3:** Table Read Protection bit[1]

1 = Block 3 (006000-007FFFh) not protected from Table Reads executed in other blocks
0 = Block 3 (006000-007FFFh) protected from Table Reads executed in other blocks

bit 2   **EBTR2:** Table Read Protection bit[1]

1 = Block 2 (004000-005FFFh) not protected from Table Reads executed in other blocks
0 = Block 2 (004000-005FFFh) protected from Table Reads executed in other blocks

bit 1   **EBTR1:** Table Read Protection bit

1 = Block 1 (002000-003FFFh) not protected from Table Reads executed in other blocks
0 = Block 1 (002000-003FFFh) protected from Table Reads executed in other blocks

bit 0   **EBTR0:** Table Read Protection bit

1 = Block 0 (000200h-001FFFh) not protected from Table Reads executed in other blocks
0 = Block 0 (000200h-001FFFh) protected from Table Reads executed in other blocks

Note 1: Unimplemented in PIC18FX42 devices; maintain this bit set.

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
- n = Value when device is unprogrammed		u = Unchanged from programmed state

REGISTER 19-11: CONFIGURATION REGISTER 7 HIGH (CONFIG7H: BYTE ADDRESS 30000Dh)

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7							bit 0

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

bit 6   **EBTRB:** Boot Block Table Read Protection bit

1 = Boot Block (000000-0001FFh) not protected from Table Reads executed in other blocks
0 = Boot Block (000000-0001FFh) protected from Table Reads executed in other blocks

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

Legend:		
R = Readable bit	C =Clearable bit	U = Unimplemented bit, read as '0'
- n = Value when device is unprogrammed		u = Unchanged from programmed state

BRA	**Unconditional Branch**

Syntax:	[*label*] BRA n
Operands:	$-1024 \leq n \leq 1023$
Operation:	$(PC) + 2 + 2n \rightarrow PC$
Status Affected:	None
Encoding:	

1101	0nnn	nnnn	nnnn

Description:	Add the 2's complement number '2n' 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 a two-cycle instruction.
Words:	1
Cycles:	2

Q Cycle Activity:

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

<u>Example</u>:   HERE   BRA   Jump

Before Instruction
    PC   =   address (HERE)
After Instruction
    PC   =   address (Jump)

BSF	**Bit Set f**

Syntax:	[*label*] BSF f,b[,a]
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$ $a \in [0,1]$
Operation:	$1 \rightarrow f<b>$
Status Affected:	None
Encoding:	

1000	bbba	ffff	ffff

Description:	Bit 'b' in register 'f' is set. If 'a' is 0 Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value.
Words:	1
Cycles:	1

Q Cycle Activity:

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

<u>Example</u>:   BSF   FLAG_REG, 7, 1

Before Instruction
    FLAG_REG   =   0x0A
After Instruction
    FLAG_REG   =   0x8A

PIC18FXX2

RLNCF	Rotate Left f (no carry)
Syntax:	[*label*] RLNCF f [,d [,a]
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f<n>) → dest<n+1>, (f<7>) → dest<0>
Status Affected:	N, Z
Encoding:	0100	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is 0, the result is placed 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).

```
  ┌─────────────────┐
  └──── register f ◄─┘
```

Words:	1
Cycles:	1

Q Cycle Activity:

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

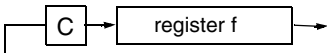Example: RLNCF   REG, 1, 0

Before Instruction
    REG   =   1010 1011
After Instruction
    REG   =   0101 0111

RRCF	Rotate Right f through Carry
Syntax:	[*label*] RRCF f [,d [,a]
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f<n>) → dest<n-1>, (f<0>) → C, (C) → dest<7>
Status Affected:	C, N, Z
Encoding:	0011	00da	ffff	ffff

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 W. If 'd' is 1, the result is placed 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).

```
  ┌───┐  ┌─────────────────┐
→ │ C ├─►│   register f    ├─►
  └───┘  └─────────────────┘
```

Words:	1
Cycles:	1

Q Cycle Activity:

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

Example: RRCF   REG, 0, 0

Before Instruction
    REG   =   1110 0110
    C   =   0
After Instruction
    REG   =   1110 0110
    W   =   0111 0011
    C   =   0

TSTFSZ   **Test f, skip if 0**

Syntax:	[*label*] TSTFSZ f [,a]
Operands:	$0 \le f \le 255$ $a \in [0,1]$
Operation:	skip if f = 0
Status Affected:	None

Encoding:

0110	011a	ffff	ffff

Description:	If 'f' = 0, the next instruction, fetched during the current instruction execution, is discarded and a NOP is executed, making this a two-cycle instruction. 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(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
         HERE    TSTFSZ CNT, 1
         NZERO   :
         ZERO    :
```

Before Instruction
    PC = Address (HERE)
After Instruction
    If CNT   =   0x00,
    PC   =   Address (ZERO)
    If CNT   ≠   0x00,
    PC   =   Address (NZERO)

XORLW   **Exclusive OR literal with W**

Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k $\rightarrow$ W
Status Affected:	N, Z

Encoding:

0000	1010	kkkk	kkkk

Description:	The contents of W are XORed with 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: XORLW 0xAF

Before Instruction
    W   =   0xB5
After Instruction
    W   =   0x1A

21.8   MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PICmicro MCUs and can be used to develop for this and other PICmicro microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

21.9   PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable V_{DD} and V_{PP} supplies, which allow it to verify programmed memory at V_{DD} min and V_{DD} max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PICmicro devices. It can also set code protection in this mode.

21.10   PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PICmicro devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

21.11   PICDEM 1 Low Cost PICmicro Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

21.12   PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I²C™ bus and separate headers for connection to an LCD module and a keypad.

TABLE 21-1: DEVELOPMENT TOOLS FROM MICROCHIP

Category	Tool	PIC12CXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X/PIC16F8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	PIC18FXXX	24CXX/25CXX/93CXX	HCSXXX	MCRFXXX	MCP2510
Software Tools	MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	MPLAB® C17 C Compiler												✓	✓						
	MPLAB® C18 C Compiler														✓	✓				
	MPASM™ Assembler/ MPLINK™ Object Linker	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Emulators	MPLAB® ICE In-Circuit Emulator	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	ICEPIC™ In-Circuit Emulator	✓			✓			✓	✓	✓		✓								
Debugger	MPLAB® ICD In-Circuit Debugger				✓*			✓*			✓					✓				
Programmers	PICSTART® Plus Entry Level Development Programmer	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	PRO MATE® II Universal Device Programmer	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Demo Boards and Eval Kits	PICDEM™ 1 Demonstration Board			✓		✓		✓†					✓							
	PICDEM™ 2 Demonstration Board				✓†			✓†							✓	✓				
	PICDEM™ 3 Demonstration Board											✓								
	PICDEM™ 14A Demonstration Board		✓																	
	PICDEM™ 17 Demonstration Board													✓						
	KEELOQ® Evaluation Kit																	✓		
	KEELOQ® Transponder Kit																	✓		
	microID™ Programmer's Kit																		✓	
	125 kHz microID™ Developer's Kit																		✓	
	125 kHz Anticollision microID™ Developer's Kit																		✓	
	13.56 MHz Anticollision microID™ Developer's Kit																		✓	
	MCP2510 CAN Developer's Kit																			✓

* Contact the Microchip Technology Inc. web site at www.microchip.com for information on how to use the MPLAB® ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77.

** Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

PIC18FXX2

TABLE 22-2:   MEMORY PROGRAMMING REQUIREMENTS

DC Characteristics			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Internal Program Memory Programming Specifications					
D110	VPP	Voltage on $\overline{MCLR}$/VPP pin	9.00	—	13.25	V	
D113	IDDP	Supply Current during Programming	—	—	10	mA	
		Data EEPROM Memory					
D120	ED	Cell 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[1]	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: Refer to Section 6.8 for a more detailed discussion on data EEPROM endurance.

FIGURE 22-11: PARALLEL SLAVE PORT TIMING (PIC18F4X2)

Note: Refer to Figure 22-4 for load conditions.
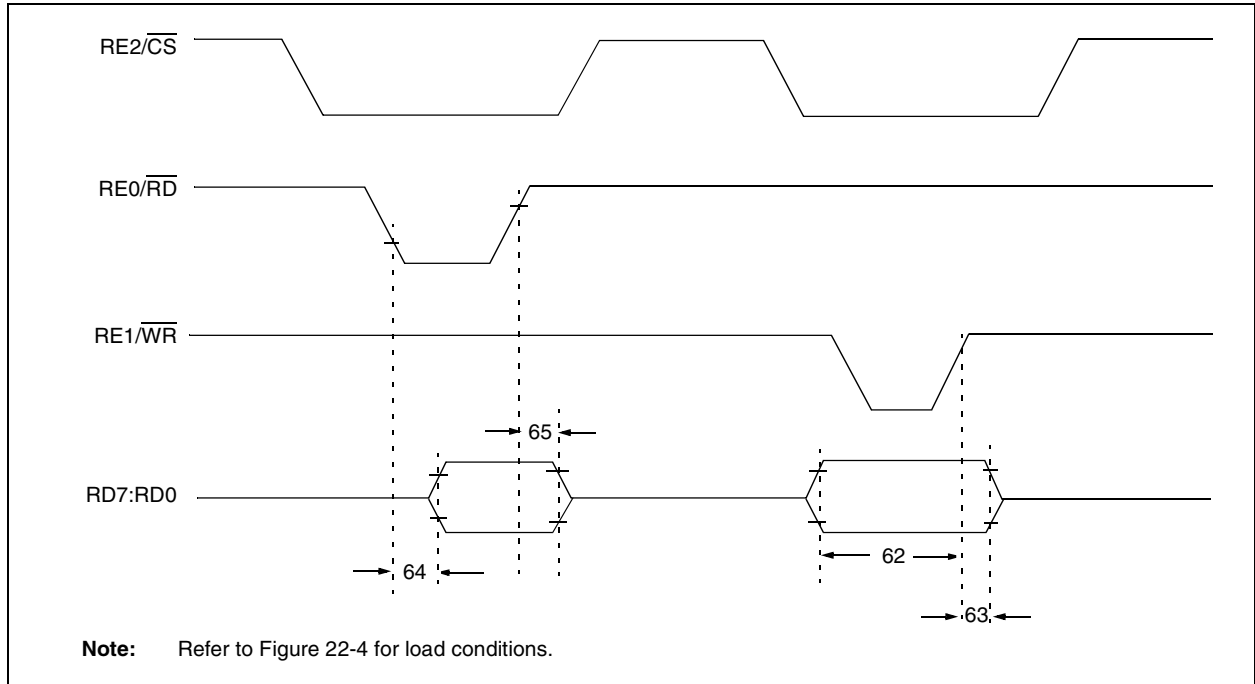
TABLE 22-10: PARALLEL SLAVE PORT REQUIREMENTS (PIC18F4X2)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
62	TdtV2wrH	Data in valid before WR↑ or CS↑ (setup time)		20	—	ns	
				25	—	ns	Extended Temp. Range
63	TwrH2dtI	WR↑ or CS↑ to data–in invalid (hold time)	PIC18**F**XXX	20	—	ns	
			PIC18**LF**XXX	35	—	ns	VDD = 2V
64	TrdL2dtV	RD↓ and CS↓ to data–out valid		—	80	ns	
				—	90	ns	Extended Temp. Range
65	TrdH2dtI	RD↑ or CS↓ to data–out invalid		10	30	ns	
66	TibfINH	Inhibit of the IBF flag bit being cleared from WR↑ or CS↑		—	3 TCY		

FIGURE 22-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

Note: Refer to Figure 22-4 for load conditions.
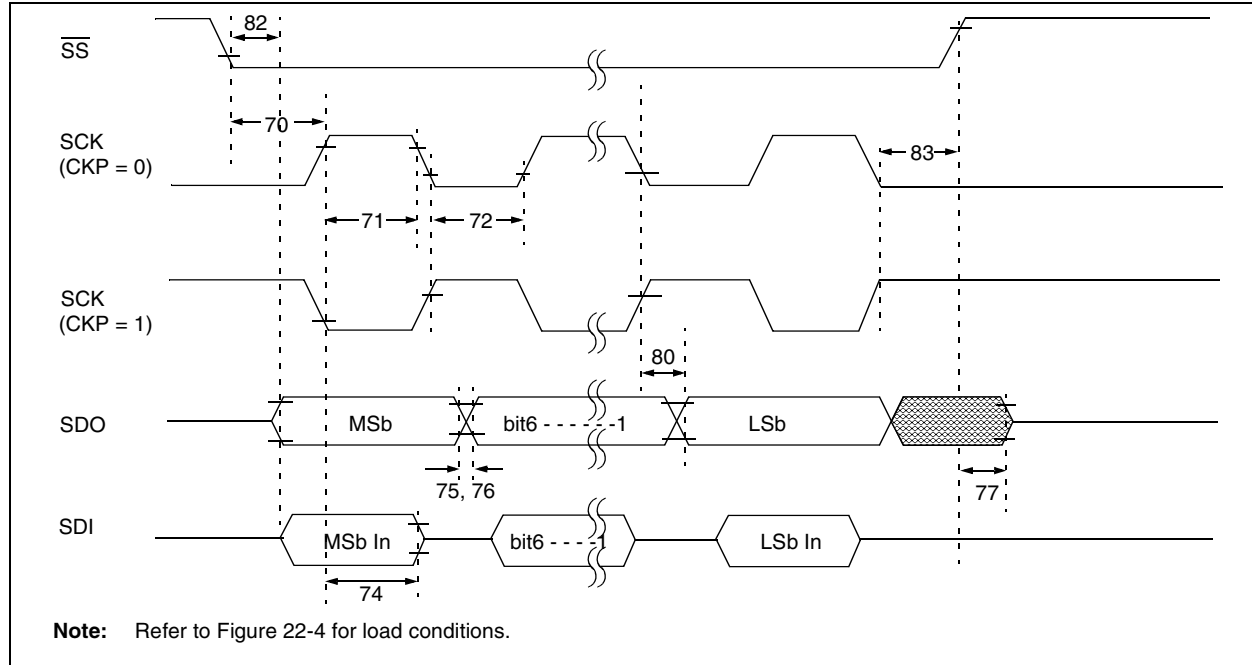
TABLE 22-14: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input		T$_{CY}$	—	ns	
71	TscH	SCK input high time (Slave mode)	Continuous	1.25 T$_{CY}$ + 30	—	ns	
71A			Single Byte	40	—	ns	**(Note 1)**
72	TscL	SCK input low time (Slave mode)	Continuous	1.25 T$_{CY}$ + 30	—	ns	
72A			Single Byte	40	—	ns	**(Note 1)**
73A	T$_{B2B}$	Last clock edge of Byte1 to the first clock edge of Byte2		1.5 T$_{CY}$ + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	ns	
75	TdoR	SDO data output rise time	PIC18**F**XXX	—	25	ns	
			PIC18**LF**XXX	—	60	ns	V$_{DD}$ = 2V
76	TdoF	SDO data output fall time	PIC18**F**XXX	—	25	ns	
			PIC18**LF**XXX	—	60	ns	V$_{DD}$ = 2V
77	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance		10	50	ns	
78	TscR	SCK output rise time (Master mode)	PIC18**F**XXX	—	25	ns	
			PIC18**LF**XXX	—	60	ns	V$_{DD}$ = 2V
79	TscF	SCK output fall time (Master mode)	PIC18**F**XXX	—	25	ns	
			PIC18**LF**XXX	—	60	ns	V$_{DD}$ = 2V
80	TscH2doV, TscL2doV	SDO data output valid after SCK edge	PIC18**F**XXX	—	50	ns	
			PIC18**LF**XXX	—	150	ns	V$_{DD}$ = 2V
82	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	PIC18**F**XXX	—	50	ns	
			PIC18**LF**XXX	—	150	ns	V$_{DD}$ = 2V
83	TscH2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge		1.5 T$_{CY}$ + 40	—	ns	

Note 1: Requires the use of Parameter # 73A.
　　2: Only if Parameter # 71A and # 72A are used.

FIGURE 23-13: **AVERAGE F**OSC **vs. V**DD **FOR VARIOUS VALUES OF R**
(RC MODE, C = 100 pF, +25°C)
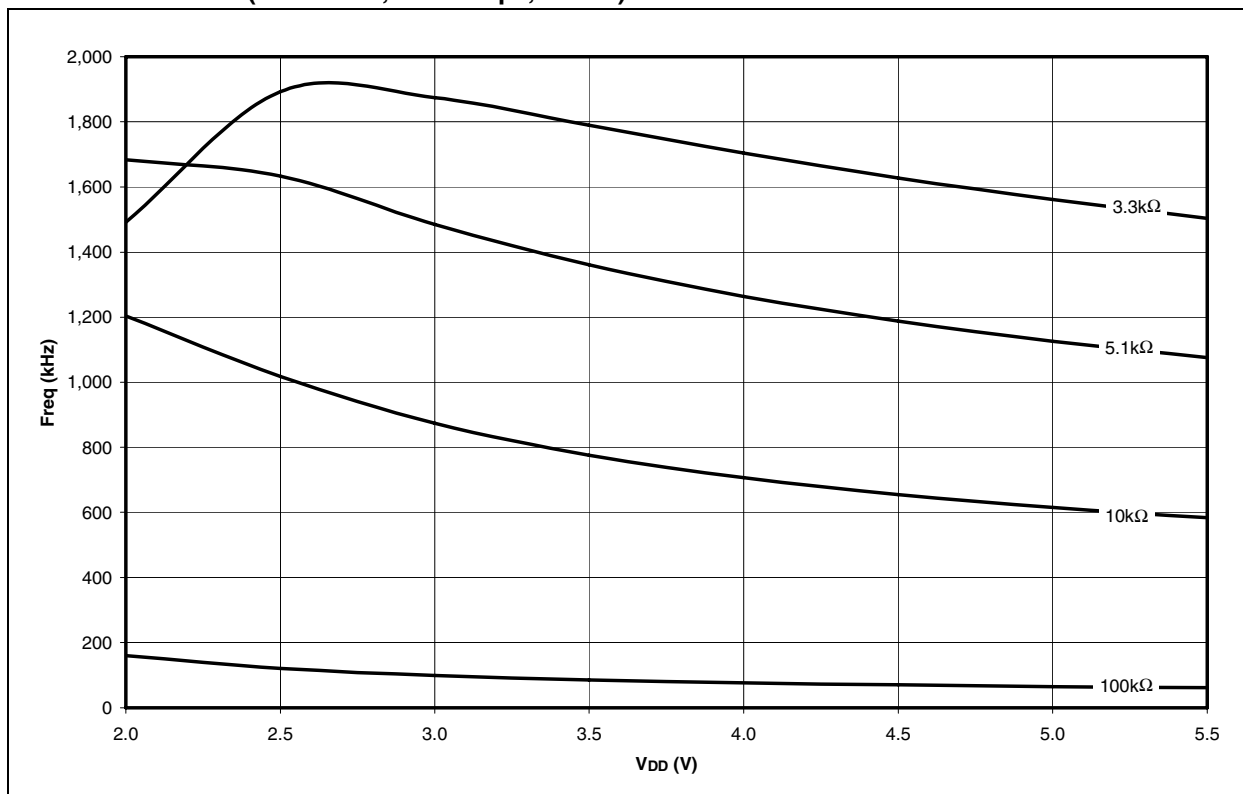
FIGURE 23-14: **AVERAGE F**OSC **vs. V**DD **FOR VARIOUS VALUES OF R**
(RC MODE, C = 300 pF, +25°C)
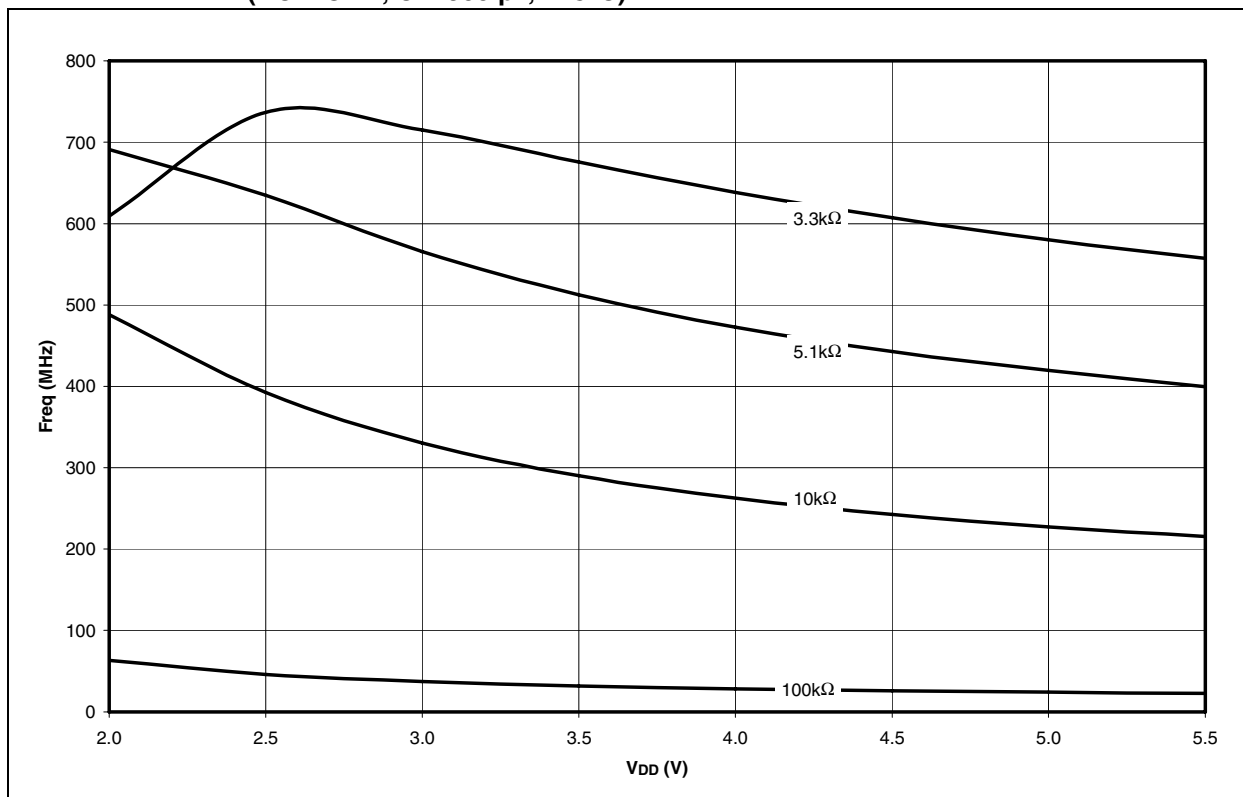
44-Lead Plastic Leaded Chip Carrier (L) – Square (PLCC)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging
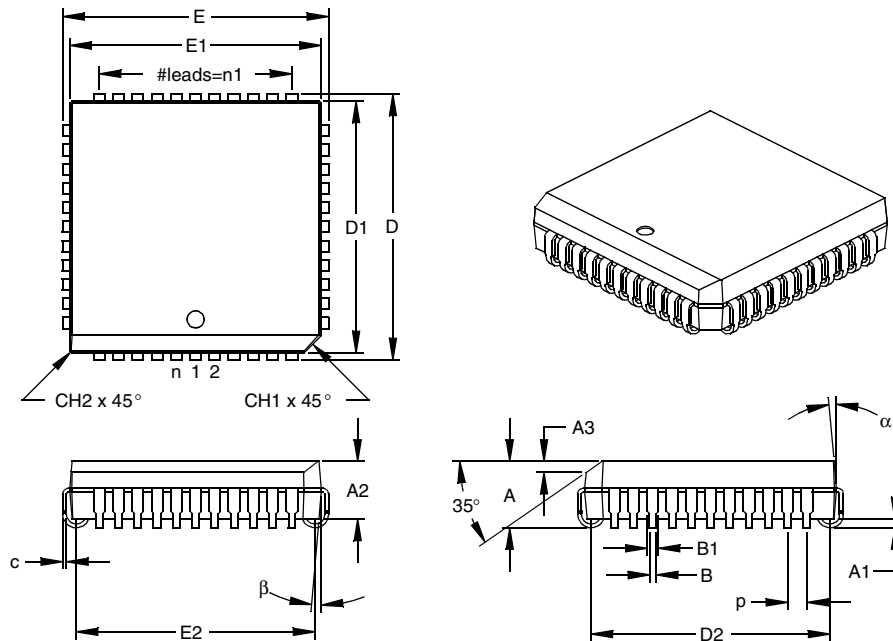
	Units	INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	p		.050			1.27	
Pins per Side	n1		11			11	
Overall Height	A	.165	.173	.180	4.19	4.39	4.57
Molded Package Thickness	A2	.145	.153	.160	3.68	3.87	4.06
Standoff §	A1	.020	.028	.035	0.51	0.71	0.89
Side 1 Chamfer Height	A3	.024	.029	.034	0.61	0.74	0.86
Corner Chamfer 1	CH1	.040	.045	.050	1.02	1.14	1.27
Corner Chamfer (others)	CH2	.000	.005	.010	0.00	0.13	0.25
Overall Width	E	.685	.690	.695	17.40	17.53	17.65
Overall Length	D	.685	.690	.695	17.40	17.53	17.65
Molded Package Width	E1	.650	.653	.656	16.51	16.59	16.66
Molded Package Length	D1	.650	.653	.656	16.51	16.59	16.66
Footprint Width	E2	.590	.620	.630	14.99	15.75	16.00
Footprint Length	D2	.590	.620	.630	14.99	15.75	16.00
Lead Thickness	c	.008	.011	.013	0.20	0.27	0.33
Upper Lead Width	B1	.026	.029	.032	0.66	0.74	0.81
Lower Lead Width	B	.013	.020	.021	0.33	0.51	0.53
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MO-047
Drawing No. C04-048

INDEX

PIC18FXX2

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager   Total Pages Sent _____

RE: Reader Response

From: Name _____

      Company _____

      Address _____

      City / State / ZIP / Country _____

      Telephone: (_____) _____ - _____   FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ____Y ____N

Device: PIC18FXX2   Literature Number: DS39564C

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?