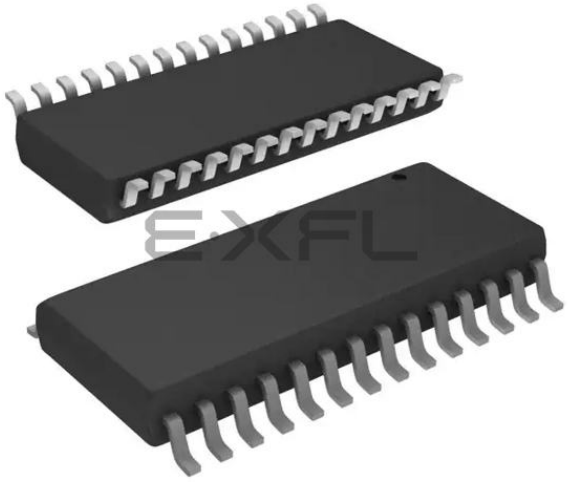


Welcome to E-XFL.COM

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	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	22
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)	4.2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f248-e-so

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, Mindi, MiWi, MPASM, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

Printed on recycled paper.

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

3.7 Time-out Sequence

On power-up, the time-out sequence is as follows: First, PWRT time-out is invoked after the POR time delay has expired, then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 3-3, Figure 3-4, Figure 3-5, Figure 3-6 and Figure 3-7 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately (Figure 3-5). This is useful for testing purposes or to synchronize more than one PIC18FXX8 device operating in parallel.

Table 3-2 shows the Reset conditions for some Special Function Registers, while Table 3-3 shows the Reset conditions for all registers.

TABLE 3-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up$^{(2)}$		Brown-out$^{(2)}$	Wake-up from Sleep or Oscillator Switch
	$\overline{\text{PWRTEN}}$ = 0	$\overline{\text{PWRTEN}}$ = 1		
HS with PLL enabled$^{(1)}$	72 ms + 1024 TOSC + 2 ms	1024 TOSC + 2 ms	72 ms + 1024 TOSC + 2 ms	1024 TOSC + 2 ms
HS, XT, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
EC	72 ms	—	72 ms	—
External RC	72 ms	—	72 ms	—

Note 1: 2 ms = Nominal time required for the 4x PLL to lock.

2: 72 ms is the nominal Power-up Timer delay.

REGISTER 3-1: RCON REGISTER BITS AND POSITIONS

R/W-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-1
IPEN	—	—	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

TABLE 3-2: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter	RCON Register	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	STKFUL	STKUNF
Power-on Reset	0000h	0--1 110q	1	1	1	0	0	u	u
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	0--0 011q	u	u	u	u	u	u	u
Software Reset during normal operation	0000h	0--0 011q	0	u	u	u	u	u	u
Stack Full Reset during normal operation	0000h	0--0 011q	u	u	u	1	1	u	1
Stack Underflow Reset during normal operation	0000h	0--0 011q	u	u	u	1	1	1	u
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	0--0 011q	u	1	0	u	u	u	u
WDT Reset	0000h	0--0 011q	u	0	1	u	u	u	u
WDT Wake-up	PC + 2	0--1 101q	u	0	0	u	u	u	u
Brown-out Reset	0000h	0--1 110q	1	1	1	u	0	u	u
Interrupt wake-up from Sleep	PC + 2$^{(1)}$	0--1 101q	u	1	0	u	u	u	u

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bits are set, the PC is loaded with the interrupt vector (000008h or 000018h).

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

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
TRISE	PIC18F2X8	PIC18F4X8	0000 -111	0000 -111	uuuu -uuu
TRISD	PIC18F2X8	PIC18F4X8	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F2X8	PIC18F4X8	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F2X8	PIC18F4X8	1111 1111	1111 1111	uuuu uuuu
TRISA(5)	PIC18F2X8	PIC18F4X8	-111 1111(5)	-111 1111(5)	-uuu uuuu(5)
LATE	PIC18F2X8	PIC18F4X8	---- -xxx	---- -uuu	---- -uuu
LATD	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA(5)	PIC18F2X8	PIC18F4X8	-xxx xxxx(5)	-uuu uuuu(5)	-uuu uuuu(5)
PORTE	PIC18F2X8	PIC18F4X8	---- -xxx	---- -000	---- -uuu
PORTD	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA(5)	PIC18F2X8	PIC18F4X8	-x0x 0000 (5)	-u0u 0000(5)	-uuu uuuu(5)
TXERRCNT	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
RXERRCNT	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
COMSTAT	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
CIOCON	PIC18F2X8	PIC18F4X8	--00 ----	--00 ----	--uu ----
BRGCON3	PIC18F2X8	PIC18F4X8	-0-- -000	-0-- -000	-u-- -uuu
BRGCON2	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
BRGCON1	PIC18F2X8	PIC18F4X8	0000 0000	0000 0000	uuuu uuuu
CANCON	PIC18F2X8	PIC18F4X8	xxxx xxx-	uuuu uuu-	uuuu uuu-
CANSTAT(6)	PIC18F2X8	PIC18F4X8	xxx- xxx-	uuu- uuu-	uuu- uuu-
RXB0D7	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D6	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D5	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D4	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D3	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D2	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D1	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D0	PIC18F2X8	PIC18F4X8	xxxx xxxx	uuuu uuuu	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 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Values for CANSTAT also apply to its other instances (CANSTATRO1 through CANSTATRO4).

4.10 Access Bank

The Access Bank is an architectural enhancement that is very useful for C compiler code optimization. The techniques used by the C compiler are also useful for programs written in assembly.

This data memory region can be used for:

- Intermediate computational values
- Local variables of subroutines
- Faster context saving/switching of variables
- Common variables
- Faster evaluation/control of SFRs (no banking)

The Access Bank is comprised of the upper 160 bytes in Bank 15 (SFRs) and the lower 96 bytes in Bank 0. These two sections will be referred to as Access Bank High and Access Bank Low, respectively. Figure 4-6 indicates the Access Bank areas.

A bit in the instruction word specifies if the operation is to occur in the bank specified by the BSR register or in the Access Bank.

When forced in the Access Bank (a = 0), the last address in Access Bank Low is followed by the first address in Access Bank High. Access Bank High maps most of the Special Function Registers so that these registers can be accessed without any software overhead.

4.11 Bank Select Register (BSR)

The need for a large general purpose memory space dictates a RAM banking scheme. The data memory is partitioned into sixteen banks. When using direct addressing, the BSR should be configured for the desired bank.

BSR<3:0> holds the upper 4 bits of the 12-bit RAM address. The BSR<7:4> bits will always read '0's and writes will have no effect.

A MOVLB instruction has been provided in the instruction set to assist in selecting banks.

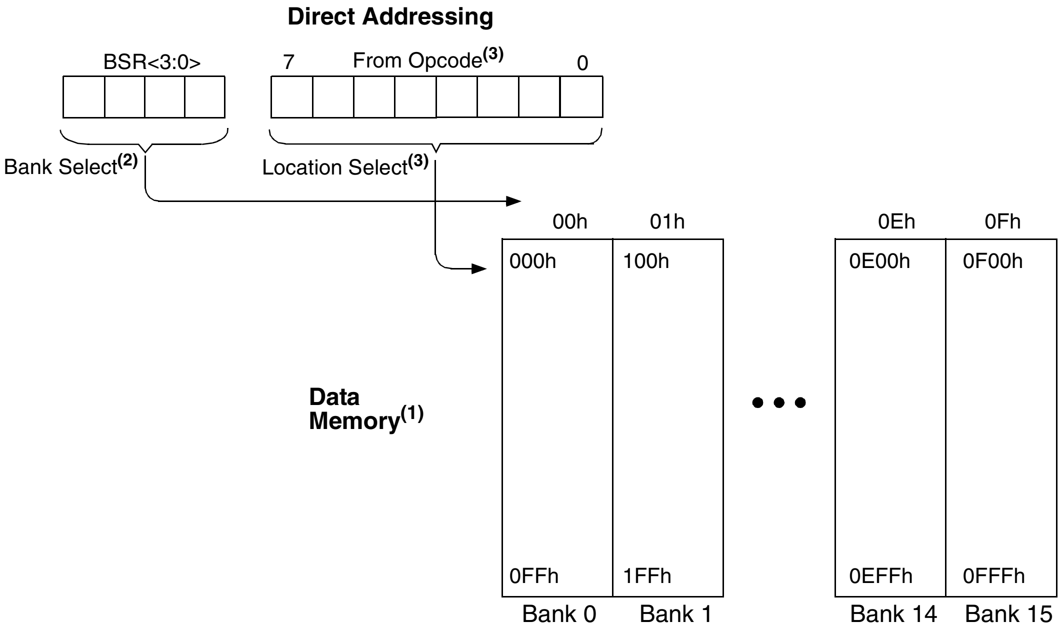
If the currently selected bank is not implemented, any read will return all '0's and all writes are ignored. The Status register bits will be set/cleared as appropriate for the instruction performed.

Each Bank extends up to FFh (256 bytes). All data memory is implemented as static RAM.

A MOVFF instruction ignores the BSR since the 12-bit addresses are embedded into the instruction word.

Section 4.12 "Indirect Addressing, INDF and FSR Registers" provides a description of indirect addressing, which allows linear addressing of the entire RAM space.

FIGURE 4-7: DIRECT ADDRESSING



Note 1: For register file map detail, see Table 4-1.

2: The access bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

3: The MOVFF instruction embeds the entire 12-bit address in the instruction.

6.5 Writing to Flash Program Memory

The minimum programming block is 4 words or 8 bytes. Word or byte programming is not supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 8 holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the `TBLWT` instruction has to be executed 8 times for each programming operation. All of the table write operations will essentially be short writes, because only the holding registers are written. At the end of updating 8 registers, the EECON1 register must be written to, to start the programming operation with a long write.

The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

6.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

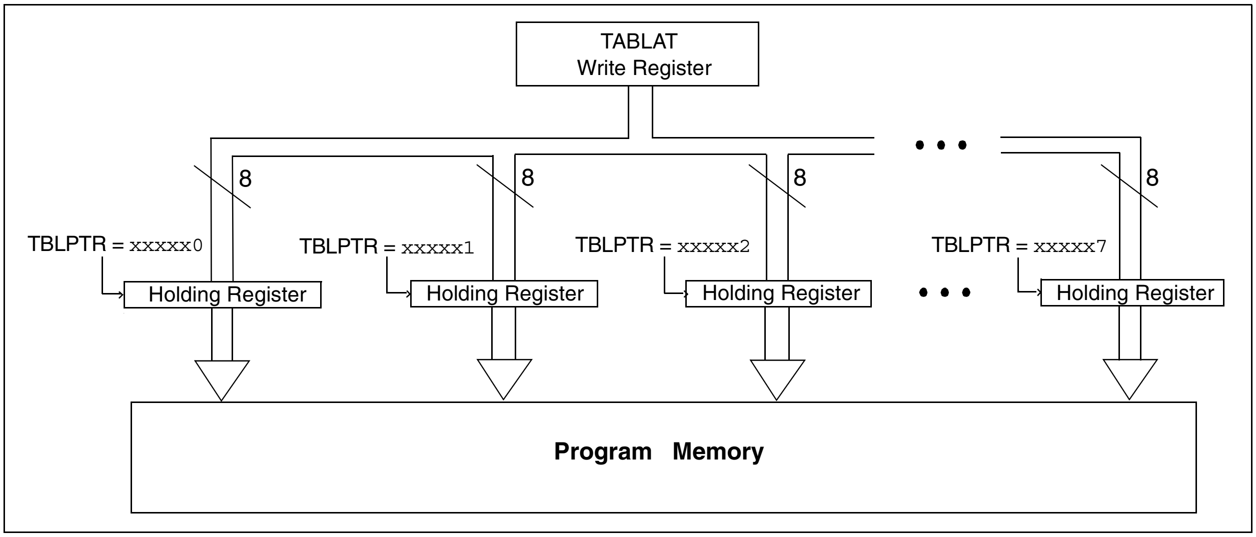
The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer with address being erased.
4. Do the row erase procedure.
5. Load Table Pointer with address of first byte being written.
6. Write the first 8 bytes into the holding registers using the `TBLWT` instruction, auto-increment may be used.
7. Set the EECON1 register for the write operation:
   - set the EEPGD bit to point to program memory;
   - clear the CFGS bit to access program memory;
   - set the WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write AAh to EECON2.
11. Set the $\overline{\text{WR}}$ bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Repeat steps 6-14 seven times to write 64 bytes.
15. Verify the memory (table read).

This procedure will require about 18 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 6-3.

> **Note:** Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 8 bytes in the holding registers.

FIGURE 6-5: TABLE WRITES TO FLASH PROGRAM MEMORY



9.4 PORTD, TRISD and LATD Registers

Note:	This port is only available on the PIC18F448 and PIC18F458.

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction register for the port is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

PORTD uses Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

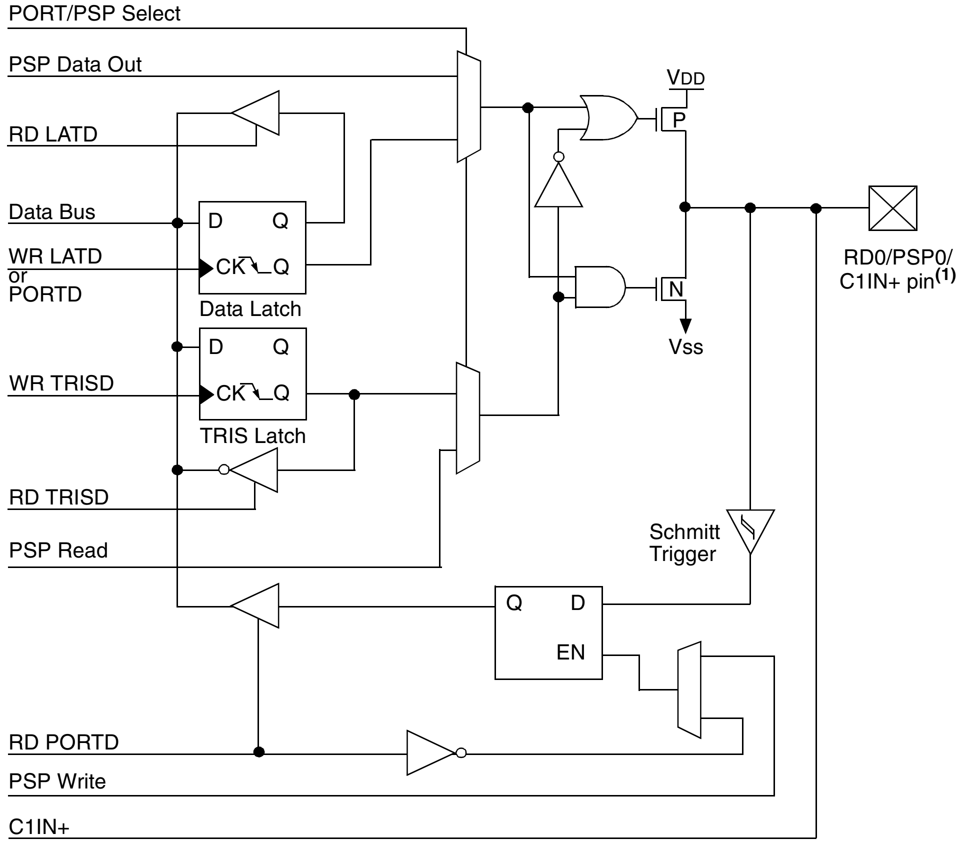
PORTD can be configured as an 8-bit wide, microprocessor port (Parallel Slave Port or PSP) by setting the control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL. See **Section 10.0 “Parallel Slave Port”** for additional information.

PORTD is also multiplexed with the analog comparator module and the ECCP module.

EXAMPLE 9-4: INITIALIZING PORTD

```
CLRF    PORTD       ; Initialize PORTD by
                    ; clearing output
                    ; data latches
CLRF    LATD        ; Alternate method
                    ; to clear output
                    ; data latches
MOVLW   07h         ; comparator off
MOVWF   CMCON
MOVLW   0CFh        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISD       ; Set RD3:RD0 as inputs
                    ; RD5:RD4 as outputs
                    ; RD7:RD6 as inputs
```

FIGURE 9-9: PORTD BLOCK DIAGRAM IN I/O PORT MODE



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

TABLE 9-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0/C1IN+	bit 0	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 0 or C1IN+ comparator input.
RD1/PSP1/C1IN-	bit 1	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 1 or C1IN- comparator input.
RD2/PSP2/C2IN+	bit 2	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 2 or C2IN+ comparator input.
RD3/PSP3/C2IN-	bit 3	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 3 or C2IN- comparator input.
RD4/PSP4/ECCP1/P1A	bit 4	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 4 or ECCP1/P1A pin.
RD5/PSP5/P1B	bit 5	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 5 or P1B pin.
RD6/PSP6/P1C	bit 6	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 6 or P1C pin.
RD7/PSP7/P1D	bit 7	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 7 or P1D pin.

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 9-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

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
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	`xxxx xxxx`	`uuuu uuuu`
LATD	LATD Data Output Register								`xxxx xxxx`	`uuuu uuuu`
TRISD	PORTD Data Direction Register								`1111 1111`	`1111 1111`
TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	`0000 -111`	`0000 -111`

Legend: `x` = unknown, `u` = unchanged, `-` = unimplemented, read as ‘`0`’. Shaded cells are not used by PORTD.

TABLE 12-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

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	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as ‘0’. Shaded cells are not used by the Timer1 module.

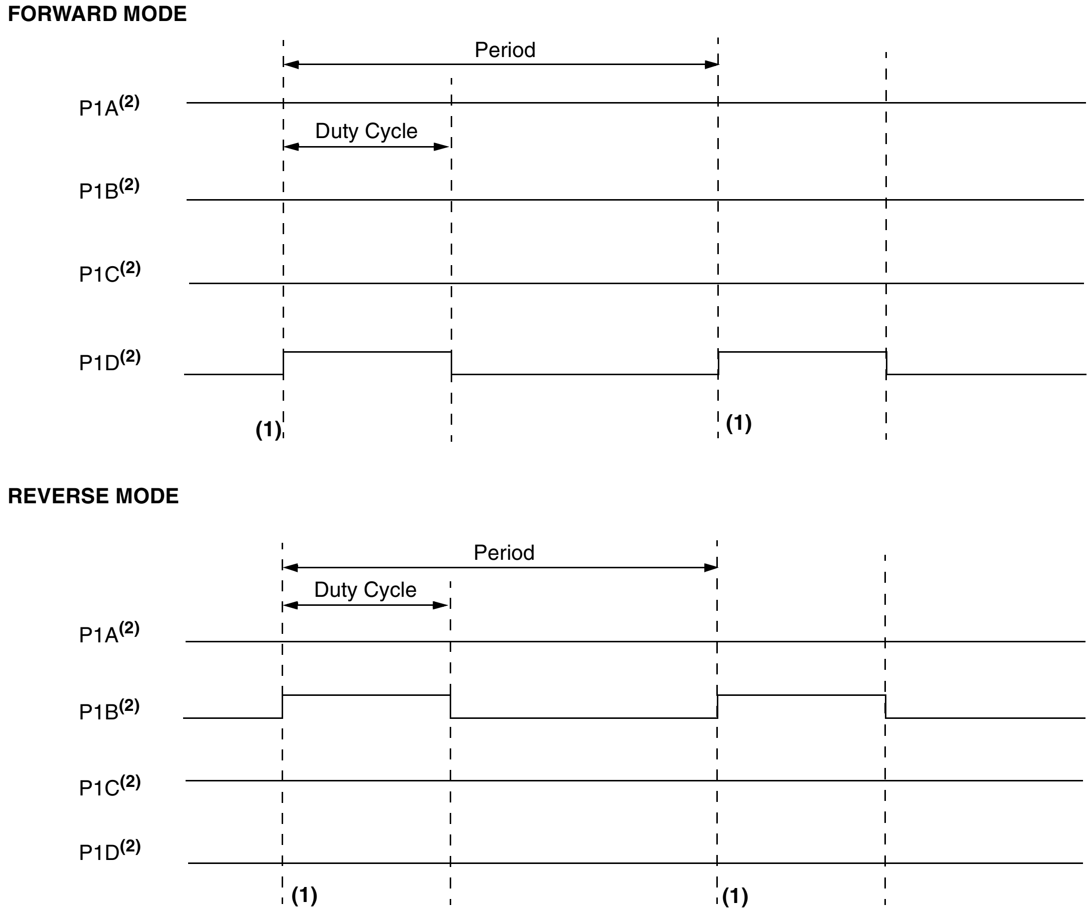
Note 1: These registers or register bits are not implemented on the PIC18F248 and PIC18F258 and read as ‘0’s.

16.5.3 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin RD4/PSP4/ECCP1/P1A is continuously active and pin RD7/PSP7/P1D is modulated. In the Reverse mode, RD6/PSP6/P1C pin is continuously active and RD5/PSP5/P1B pin is modulated. These are illustrated in Figure 16-5.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORTD<4:7> data latches. The TRISD<4:7> bits must be cleared to make the P1A, P1B, P1C and P1D pins output.

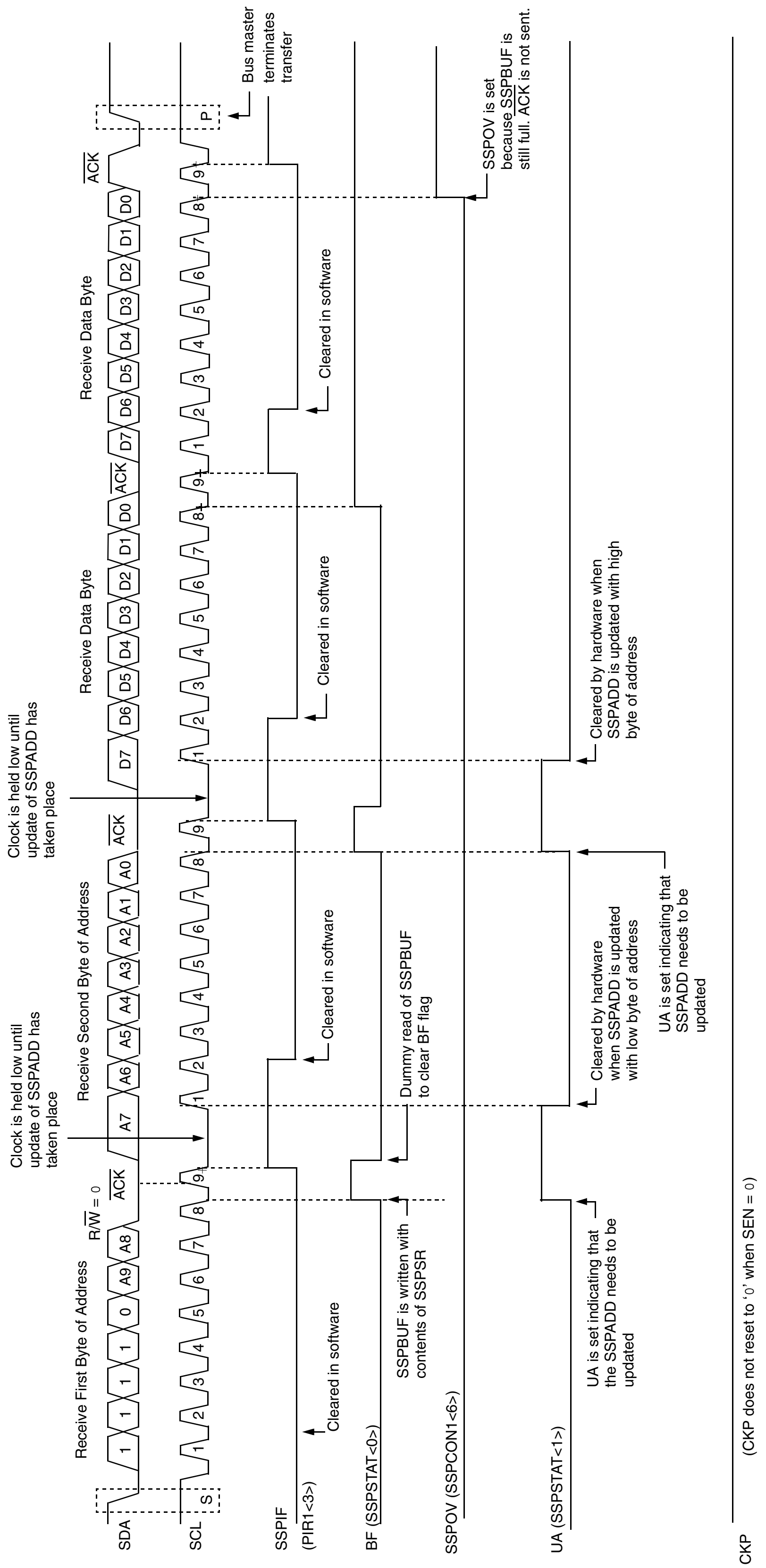
FIGURE 16-5: FULL-BRIDGE PWM OUTPUT



Note 1: At this time, the TMR2 register is equal to the PR2 register.

2: Output signal is shown as asserted high.

FIGURE 17-10: I^2C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)


SDA
SCL
SSPIF
(PIR1<3>)
BF (SSPSTAT<0>)
SSPOV (SSPCON1<6>)
UA (SSPSTAT<1>)
CKP
S
P
Receive First Byte of Address
1
1
1
1
0
A9
A8
R/W = 0
ACK
1
2
3
4
5
6
7
8
9
Cleared in software
SSPBUF is written with contents of SSPSR
UA is set indicating that the SSPADD needs to be updated
(CKP does not reset to '0' when SEN = 0)
Clock is held low until update of SSPADD has taken place
Receive Second Byte of Address
A7
A6
A5
A4
A3
A2
A1
A0
ACK
Cleared in software
Dummy read of SSPBUF to clear BF flag
Cleared by hardware when SSPADD is updated with low byte of address
UA is set indicating that SSPADD needs to be updated
Clock is held low until update of SSPADD has taken place
Receive Data Byte
D7
D6
D5
D4
D3
D2
D1
D0
ACK
Cleared in software
Cleared by hardware when SSPADD is updated with high byte of address
Receive Data Byte
D7
D6
D5
D4
D3
D2
D1
D0
ACK
Cleared in software
SSPOV is set because SSPBUF is still full. ACK is not sent.
Bus master terminates transfer

17.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^{2}C bus may be taken when the P bit is set or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I^{2}C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

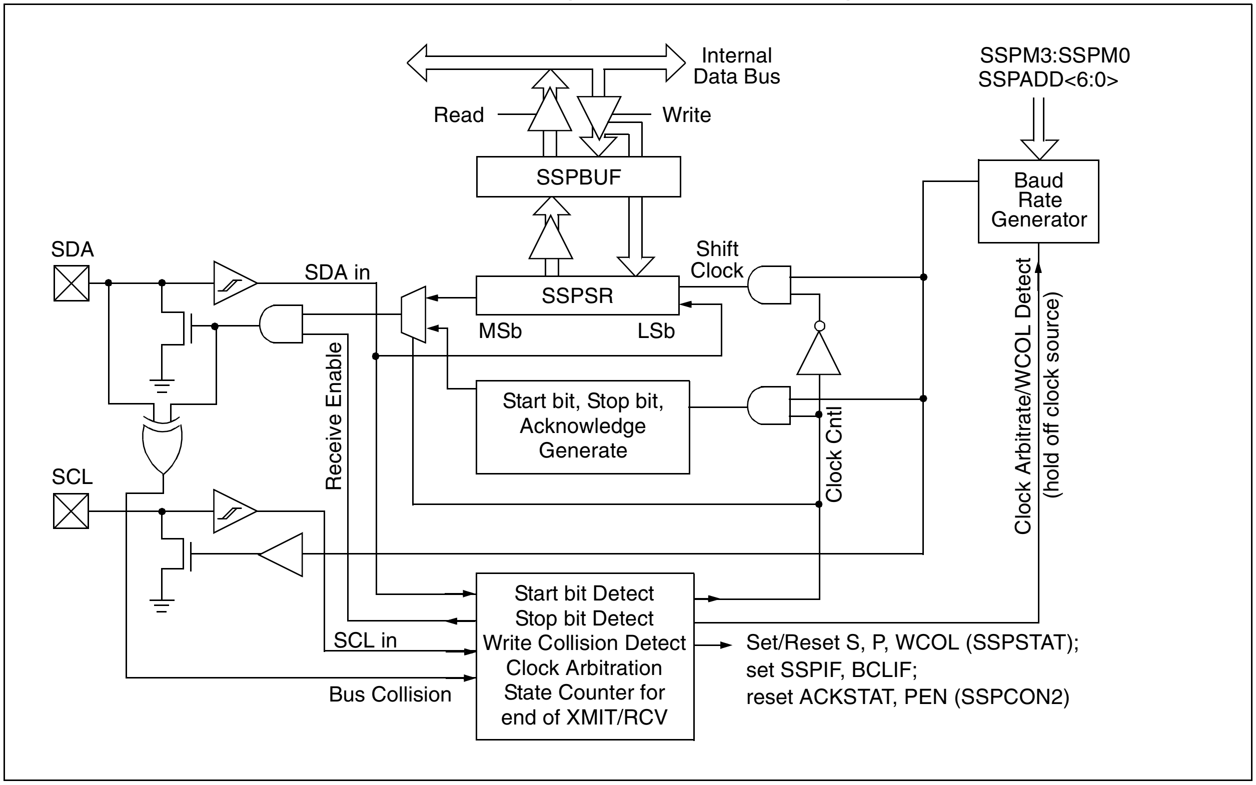
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Configure the I^{2}C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

Note:	The MSSP module, when configured in I^{2}C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start

FIGURE 17-16: MSSP BLOCK DIAGRAM (I^{2}C™ MASTER MODE)



EXAMPLE 19-1: WIN AND ICODE BITS USAGE IN INTERRUPT SERVICE ROUTINE TO ACCESS TX/RX BUFFERS (CONTINUED)

```
RXB0Interrupt
    BCF     PIR3, RXB0IF                  ; Clear the interrupt flag
    GOTO    AccessBuffer

AccessBuffer                              ; This is either TX or RX interrupt
    ; Copy CANCON.ICODE bits to CANSTAT.WIN bits
    MOVF    CANCON, W                     ; Clear CANCON.WIN bits before copying
                                          ; new ones.
    ANDLW   b’11110001’                   ; Use previously saved CANCON value to
                                          ; make sure same value.
    MOVWF   CANCON                        ; Copy masked value back to TempCANCON

    MOVF    TempCANSTAT, W                ; Retrieve ICODE bits
    ANDLW   b’00001110’                   ; Use previously saved CANSTAT value
                                          ; to make sure same value.

    IORWF   CANCON                        ; Copy ICODE bits to WIN bits.
                                          ; Copy the result to actual CANCON

    ; Access current buffer…
    ; User code

    ; Restore CANCON.WIN bits
    MOVF    CANCON, W                     ; Preserve current non WIN bits
    ANDLW   b’11110001’
    IORWF   TempCANCON, W                 ; Restore original WIN bits

    MOVWF   CANCON

    ; Do not need to restore CANSTAT - it is read-only register.

    ; Return from interrupt or check for another module interrupt source
```

REGISTER 19-3: COMSTAT: COMMUNICATION STATUS REGISTER

R/C-0	R/C-0	R-0	R-0	R-0	R-0	R-0	R-0
RXB0OVFL	RXB1OVFL	TXBO	TXBP	RXBP	TXWARN	RXWARN	EWARN
bit 7							bit 0

bit 7 **RXB0OVFL:** Receive Buffer 0 Overflow bit
1 = Receive Buffer 0 overflowed
0 = Receive Buffer 0 has not overflowed

bit 6 **RXB1OVFL:** Receive Buffer 1 Overflow bit
1 = Receive Buffer 1 overflowed
0 = Receive Buffer 1 has not overflowed

bit 5 **TXBO:** Transmitter Bus-Off bit
1 = Transmit Error Counter > 255
0 = Transmit Error Counter ≤ 255

bit 4 **TXBP:** Transmitter Bus Passive bit
1 = Transmission Error Counter > 127
0 = Transmission Error Counter ≤ 127

bit 3 **RXBP:** Receiver Bus Passive bit
1 = Receive Error Counter > 127
0 = Receive Error Counter ≤ 127

bit 2 **TXWARN:** Transmitter Warning bit
1 = 127 ≥ Transmit Error Counter > 95
0 = Transmit Error Counter ≤ 95

bit 1 **RXWARN:** Receiver Warning bit
1 = 127 ≥ Receive Error Counter > 95
0 = Receive Error Counter ≤ 95

bit 0 **EWARN:** Error Warning bit
This bit is a flag of the RXWARN and TXWARN bits.
1 = The RXWARN or the TXWARN bits are set
0 = Neither the RXWARN or the TXWARN bits are set

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

19.2.4 CAN BAUD RATE REGISTERS

This subsection describes the CAN Baud Rate registers.

REGISTER 19-29: BRGCON1: BAUD RATE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
bit 7							bit 0

bit 7-6 **SJW1:SJW0:** Synchronized Jump Width bits

`11` = Synchronization Jump Width Time = 4 x TQ
`10` = Synchronization Jump Width Time = 3 x TQ
`01` = Synchronization Jump Width Time = 2 x TQ
`00` = Synchronization Jump Width Time = 1 x TQ

bit 5-0 **BRP5:BRP0:** Baud Rate Prescaler bits

`111111` = TQ = (2 x 64)/FOSC
`111110` = TQ = (2 x 63)/FOSC
:
:
`000001` = TQ = (2 x 2)/FOSC
`000000` = TQ = (2 x 1)/FOSC

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

Note: This register is accessible in Configuration mode only.

19.6 Message Acceptance Filters and Masks

The message acceptance filters and masks are used to determine if a message in the message assembly buffer should be loaded into either of the receive buffers. Once a valid message has been received into the MAB, the identifier fields of the message are compared to the filter values. If there is a match, that message will be loaded into the appropriate receive buffer. The filter masks are used to determine which bits in the identifier are examined with the filters. A truth table is shown below in Table 19-2 that indicates how each bit in the identifier is compared to the masks and filters to determine if a message should be loaded into a receive buffer. The mask essentially determines which bits to apply the acceptance filters to. If any mask bit is set to a zero, then that bit will automatically be accepted regardless of the filter bit.

TABLE 19-2: FILTER/MASK TRUTH TABLE

Mask bit n	Filter bit n	Message Identifier bit n001	Accept or Reject bit n
0	x	x	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

Legend: x = don't care

As shown in the receive buffer block diagram (Figure 19-4), acceptance filters RXF0 and RXF1 and filter mask RXM0 are associated with RXB0. Filters RXF2, RXF3, RXF4 and RXF5 and mask RXM1 are associated with RXB1. When a filter matches and a message is loaded into the receive buffer, the filter number that enabled the message reception is loaded into the FILHIT bit(s).

For RXB1, the RXB1CON register contains the FILHIT<2:0> bits. They are coded as follows:

- 101 = Acceptance Filter 5 (RXF5)
- 100 = Acceptance Filter 4 (RXF4)
- 011 = Acceptance Filter 3 (RXF3)
- 010 = Acceptance Filter 2 (RXF2)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0 (RXF0)

Note:	'000' and '001' can only occur if the RXB0DBEN bit is set in the RXB0CON register allowing RXB0 messages to rollover into RXB1.

The coding of the RXB0DBEN bit enables these three bits to be used similarly to the FILHIT bits and to distinguish a hit on filter RXF0 and RXF1, in either RXB0, or after a rollover into RXB1.

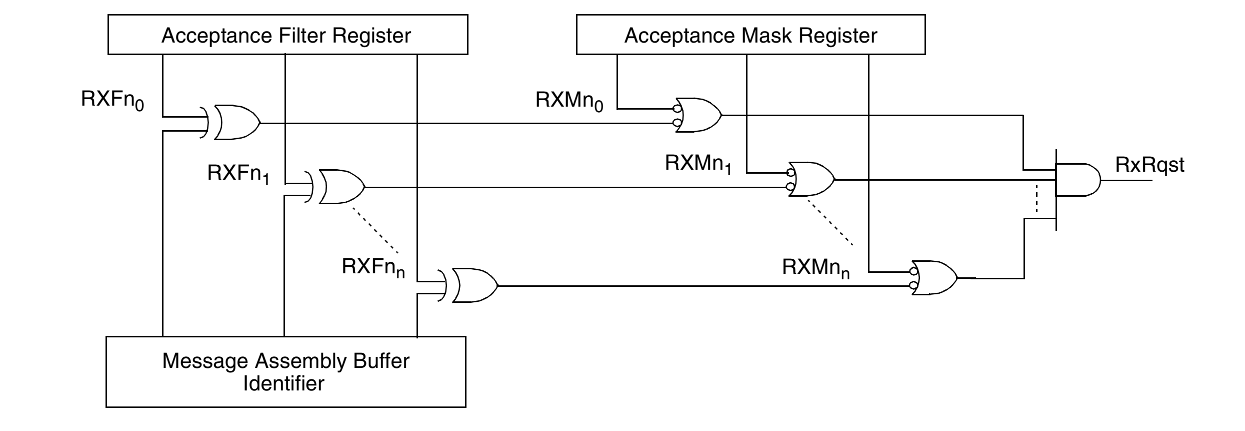
- 111 = Acceptance Filter 1 (RXF1)
- 110 = Acceptance Filter 0 (RXF0)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0

If the RXB0DBEN bit is clear, there are six codes corresponding to the six filters. If the RXB0DBEN bit is set, there are six codes corresponding to the six filters plus two additional codes corresponding to RXF0 and RXF1 filters that rollover into RXB1.

If more than one acceptance filter matches, the FILHIT bits will encode the binary value of the lowest numbered filter that matched. In other words, if filter RXF2 and filter RXF4 match, FILHIT will be loaded with the value for RXF2. This essentially prioritizes the acceptance filters with a lower number filter having higher priority. Messages are compared to filters in ascending order of filter number.

The mask and filter registers can only be modified when the PIC18FXX8 is in Configuration mode. The mask and filter registers cannot be read outside of Configuration mode. When outside of Configuration mode, all mask and filter registers will be read as '0'.

FIGURE 19-6: MESSAGE ACCEPTANCE MASK AND FILTER OPERATION


Acceptance Filter Register
Acceptance Mask Register
RXFn0
RXMn0
RXFn1
RXMn1
RxRqst
RXFnn
RXMnn
Message Assembly Buffer Identifier

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and VSS) or the voltage level on the RA3/AN3/VREF+ pin and RA2/AN2/VREF- pin.

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.

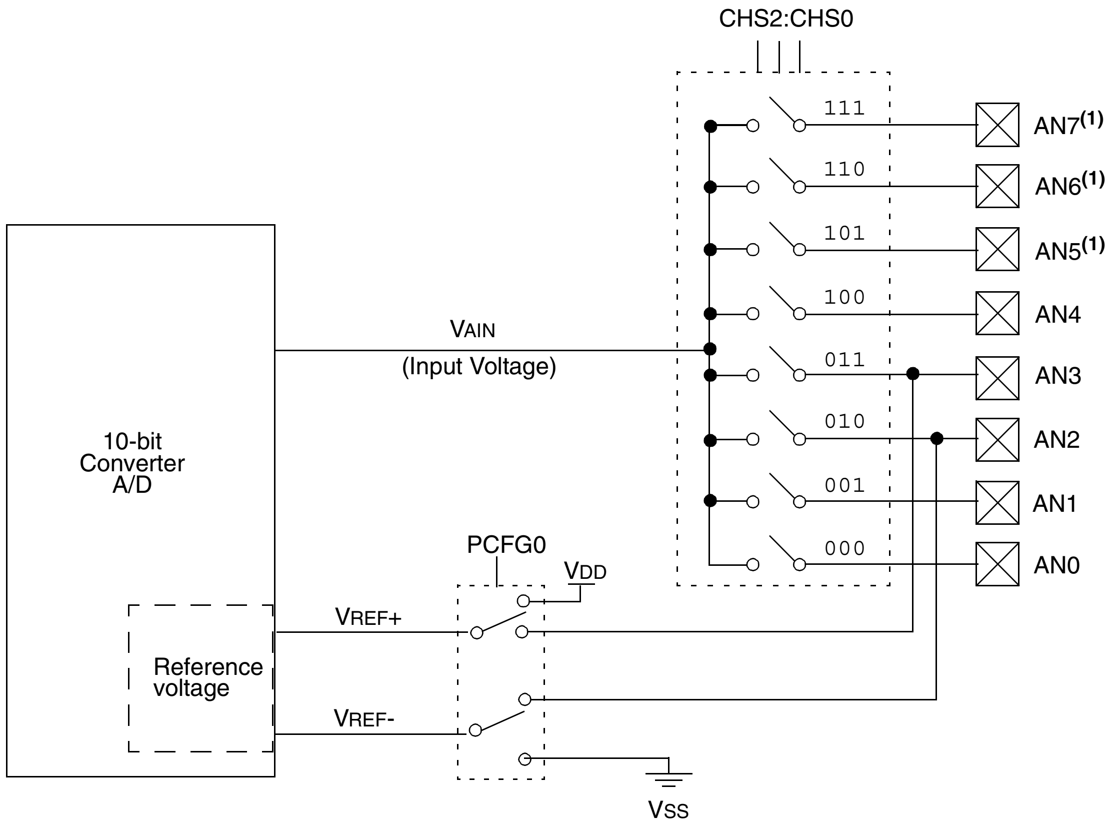
The output of the sample and hold is the input into the converter which generates the result via successive approximation.

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

Each port pin associated with the A/D converter can be configured as an analog input (RA3 can also be a voltage reference) 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/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 20-1.

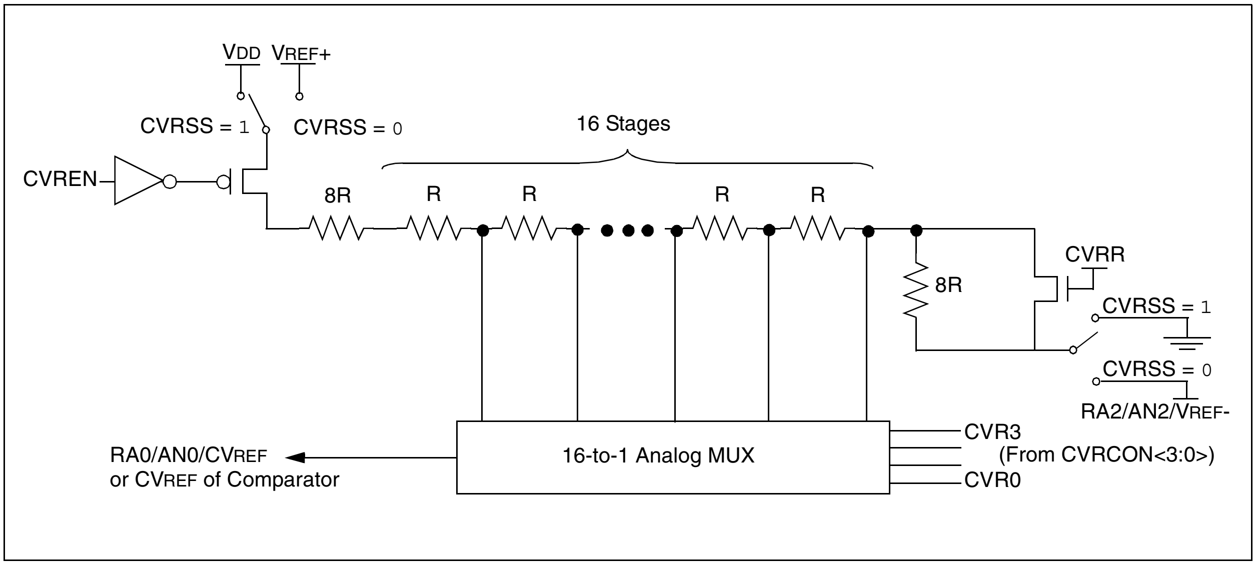
FIGURE 20-1: A/D BLOCK DIAGRAM



Note 1: Channels AN5 through AN7 are not available on PIC18F2X8 devices.

2: All I/O pins have diode protection to VDD and VSS.

FIGURE 22-1: VOLTAGE REFERENCE BLOCK DIAGRAM



22.2 Voltage Reference Accuracy/Error

The full range of voltage reference cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 22-1) keep VREF from approaching the reference source rails. The voltage reference is derived from the reference source; therefore, the VREF output changes with fluctuations in that source. The absolute accuracy of the voltage reference can be found in **Section 27.0 “Electrical Characteristics”**.

22.3 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the CVRCON register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

22.4 Effects of a Reset

A device Reset disables the voltage reference by clearing bit CVREN (CVRCON register). This Reset also disconnects the reference from the RA2 pin by clearing bit CVROE (CVRCON register) and selects the high-voltage range by clearing bit CVRR (CVRCON register). The CVRSS value select bits, CVRCON<3:0>, are also cleared.

22.5 Connection Considerations

The voltage reference module operates independently of the comparator module. The output of the reference generator may be connected to the RA0/AN0 pin if the TRISA<0> bit is set and the CVROE bit (CVRCON<6>) is set. Enabling the voltage reference output onto the RA0/AN0 pin, with an input signal present, will increase current consumption. Connecting RA0/AN0 as a digital output, with CVRSS enabled, will also increase current consumption.

The RA0/AN0 pin can be used as a simple D/A output with limited drive capability. Due to the limited current drive capability, a buffer must be used on the voltage reference output for external connections to VREF. Figure 22-2 shows an example buffering technique.

RETURN Return from Subroutine

Syntax: [*label*] RETURN [s]

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
if s = 1
(WS) → W,
(STATUSS) → Status,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	0000	0001	001s

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, Status and BSR. If 's' = 0, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Pop PC from stack
No operation	No operation	No operation	No operation

Example: RETURN

After Interrupt
PC = TOS

RLCF Rotate Left f through Carry

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

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

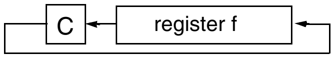
Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

Encoding:

0011	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. 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' = 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: RLCF REG, W

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

TABLE 27-3: COMPARATOR SPECIFICATIONS

Operating Conditions: V_{DD} range as described in **Section 27.1 “DC Characteristics”**, -40°C < T_A < +125°C							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	V_{IOFF}	Input Offset Voltage	—	±5.0	±10	mV	
D301	V_{ICM}	Input Common Mode Voltage	0	—	V_{DD} – 1.5	V	
D302	CMRR	CMRR	+55*	—	—	db	
D300	T_{RESP}	Response Time[(1)]	—	300* 350*	400* 600*	ns ns	PIC18FXX8 PIC18LFXX8
D301	T_{MC2OV}	Comparator Mode Change to Output Valid	—	—	10*	µs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (V_{DD} – 1.5)/2 while the other input transitions from V_{SS} to V_{DD}.

TABLE 27-4: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: V_{DD} range as described in **Section 27.1 “DC Characteristics”**, -40°C < T_A < +125°C							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D310	V_{RES}	Resolution	V_{DD}/24	—	V_{DD}/32	LSB	
D311	V_{RAA}	Absolute Accuracy	—	—	0.5	LSB	
D312	V_{RUR}	Unit Resistor Value (R)	—	2K*	—	Ω	
D310	T_{SET}	Settling Time[(1)]	—	—	10*	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while CVRR = `1` and CVR<3:0> transitions from `0000` to `1111`.