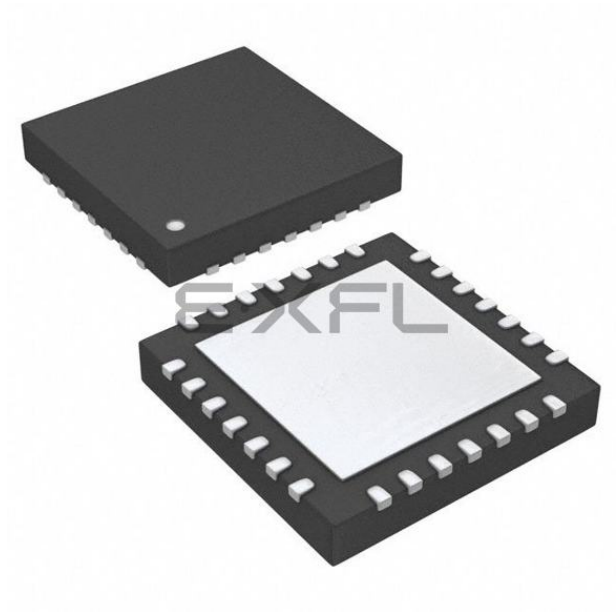


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

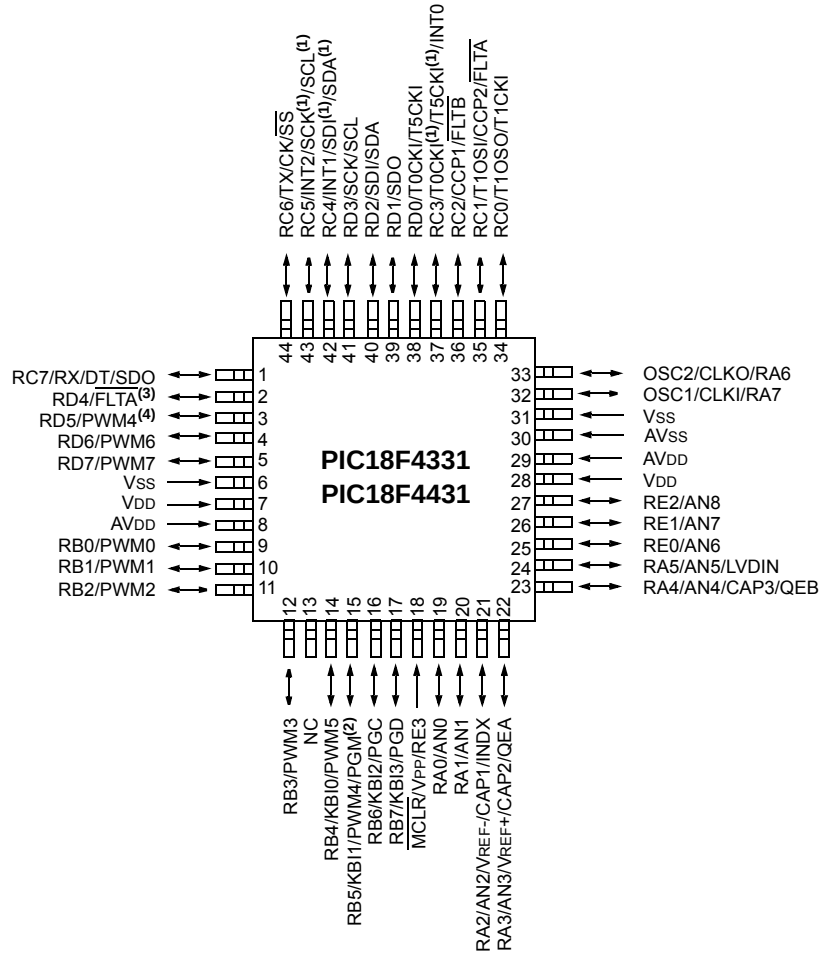
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Power Control PWM, QEI, POR, PWM, WDT
Number of I/O	24
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	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2431t-i-ml

Pin Diagrams (Continued)

44-Pin QFN(2)



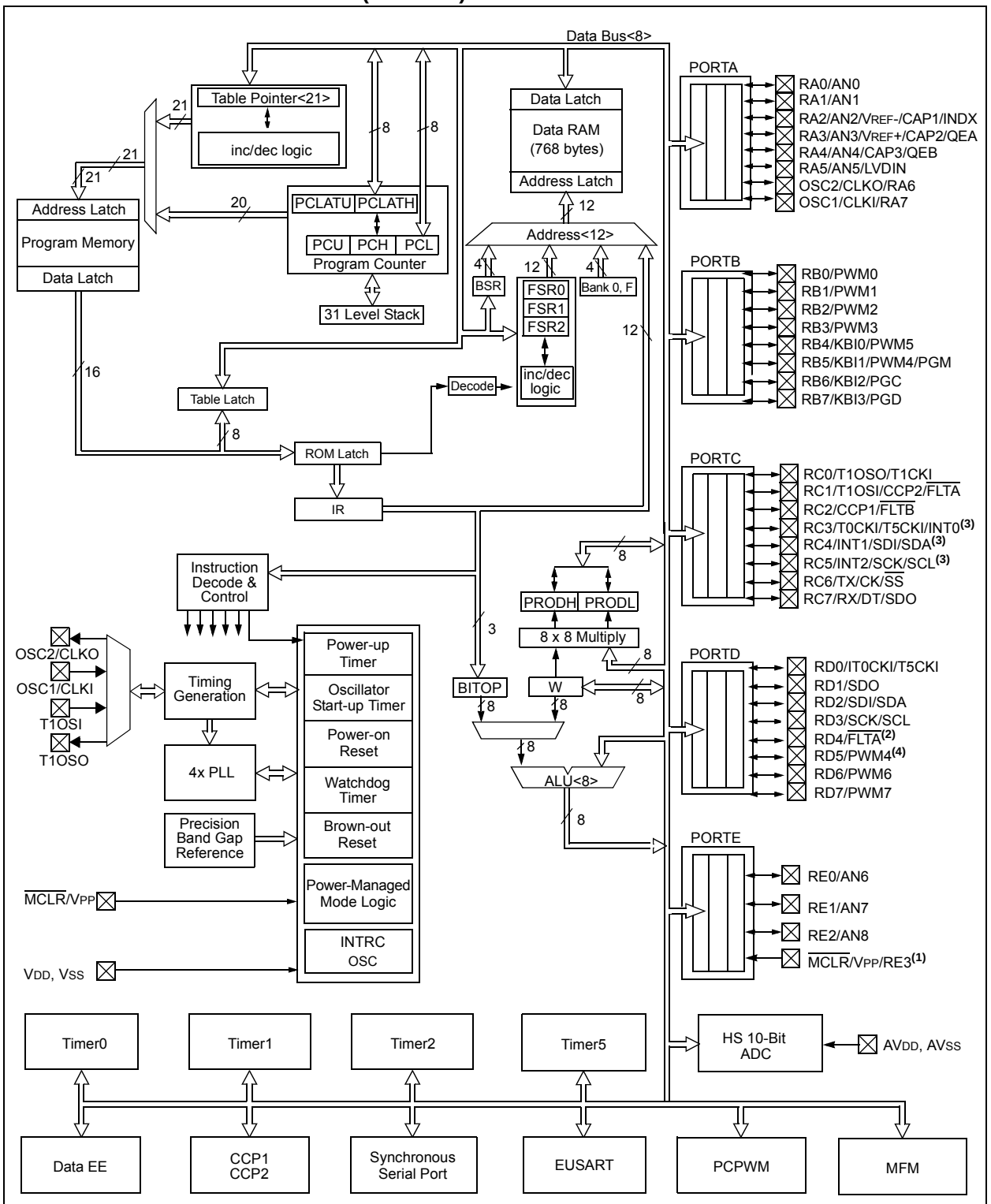
Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL.

2: For the QFN package, it is recommended that the bottom pad be connected to VSS.

3: RD4 is the alternate pin for FLTA.

4: RD5 is the alternate pin for PWM4.

FIGURE 1-2: PIC18F4331/4431 (40/44-PIN) BLOCK DIAGRAM



Note 1: RE3 is available only when $\overline{\text{MCLR}}$ is disabled.

2: RD4 is the alternate pin for $\overline{\text{FLTA}}$.

3: RC3, RC4 and RC5 are alternate pins for T0CKI/T5CKI, SDI/SDA, SCK/SCL, respectively.

4: RD5 is the alternate pin for PWM4.

TABLE 1-3: PIC18F4331/4431 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
						PORTD is a bidirectional I/O port.
RD0/T0CKI/T5CKI	19	38	38			
RD0				I/O	ST	Digital I/O.
T0CKI				I	ST	Timer0 external clock input.
T5CKI				I	ST	Timer5 input clock.
RD1/SDO	20	39	39			
RD1				I/O	ST	Digital I/O.
SDO(1)				O	—	SPI data out.
RD2/SDI/SDA	21	40	40			
RD2				I/O	ST	Digital I/O.
SDI(1)				I	ST	SPI data in.
SDA(1)				I/O	ST	I^2C™ data I/O.
RD3/SCK/SCL	22	41	41			
RD3				I/O	ST	Digital I/O.
SCK(1)				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL(1)				I/O	ST	Synchronous serial clock input/output for I^2C mode.
RD4/$\overline{\text{FLTA}}$	27	2	2			
RD4				I/O	ST	Digital I/O.
$\overline{\text{FLTA}}$(2)				I	ST	Fault interrupt input pin.
RD5/PWM4	28	3	3			
RD5				I/O	ST	Digital I/O.
PWM4(3)				O	TTL	PWM Output 4.
RD6/PWM6	29	4	4			
RD6				I/O	ST	Digital I/O.
PWM6				O	TTL	PWM Output 6.
RD7/PWM7	30	5	5			
RD7				I/O	ST	Digital I/O.
PWM7				O	TTL	PWM Output 7.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL; RC7 is the alternate pin for SDO.

2: RD4 is the alternate pin for $\overline{\text{FLTA}}$.

3: RD5 is the alternate pin for PWM4.

TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
PTCON0	2331	2431	4331	4431	0000 0000	uuuu uuuu	uuuu uuuu
PTCON1	2331	2431	4331	4431	00-- ----	00-- ----	uu-- ----
PTMRL	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PTMRH	2331	2431	4331	4431	---- 0000	---- 0000	---- uuuu
PTPERL	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
PTPERH	2331	2431	4331	4431	---- 1111	---- 1111	---- uuuu
PDC0L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC0H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
PDC1L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC1H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
PDC2L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC2H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
PDC3L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC3H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
SEVTCMPL	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SEVTCMPH	2331	2431	4331	4431	---- 0000	---- 0000	---- uuuu
PWMCON0	2331	2431	4331	4431	-111 0000	-111 0000	-uuu uuuu
PWMCON1	2331	2431	4331	4431	0000 0-00	0000 0-00	uuuu u-uu
DTCON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
FLTCONFIG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
OVDCOND	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
OVDCONS	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
CAP1BUFH/ VELRH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP1BUFL/ VELRL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP2BUFH/ POSCNTH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP2BUFL/ POSCNTL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP3BUFH/ MAXCNTH	2331	2431	4331	4431	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 5-2 for Reset value for specific condition.

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

6: Bit 3 of PORTE and LATE are enabled if $\overline{\text{MCLR}}$ functionality is disabled. When not enabled as the PORTE pin, they are disabled and read as '0'. The 28-pin devices do not have only RE3 implemented.

8.5.1 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 (see **Section 8.4.1 "Flash Program Memory Erase Sequence"**).
5. Load Table Pointer with the address of the first byte being written.
6. Write the first 8 bytes into the holding registers with auto-increment.
7. Set the EECON1 register for the write operation by doing the following:
   - Set the EEPGD bit to point to program memory
   - Clear the CFGS bit to access program memory
   - Set the WREN bit to enable byte writes
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for the duration of the write (about 2 ms using internal timer).
13. Execute a NOP.
14. Re-enable interrupts.
15. Repeat Steps 6-14 seven times to write 64 bytes.
16. 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 8-3.

11.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

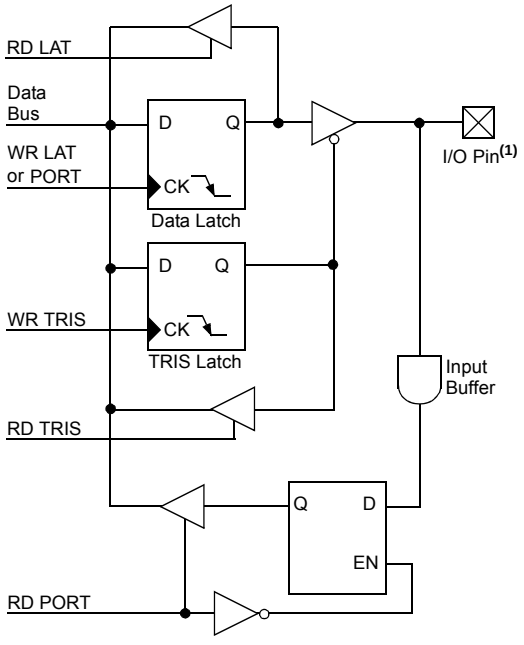
Each port has three registers for its operation. These registers are:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Data Latch)

The Data Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

A simplified model of a generic I/O port without the interfaces to other peripherals is shown in Figure 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



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

11.1 PORTA, TRISA and LATA Registers

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

Reading the PORTA register reads the status of the pins, whereas writing to it, will write to the port latch.

The Data Latch register (LATA) is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

The RA<4:2> pins are multiplexed with three input capture pins and Quadrature Encoder Interface pins. Pins, RA6 and RA7, are multiplexed with the main oscillator pins. They are enabled as oscillator or I/O pins by the selection of the main oscillator in Configuration Register 1H (see **Section 23.1 "Configuration Bits"** for details). When they are not used as port pins, RA6 and RA7 and their associated TRIS and LAT bits are read as '0'.

The other PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the comparator voltage reference output. The operation of pins RA<3:0> and RA5 as A/D Converter inputs is selected by clearing/setting the control bits in the ANSEL0 and ANSEL1 registers.

Note 1: On a Power-on Reset, RA<5:0> are configured as analog inputs and read as '0'.

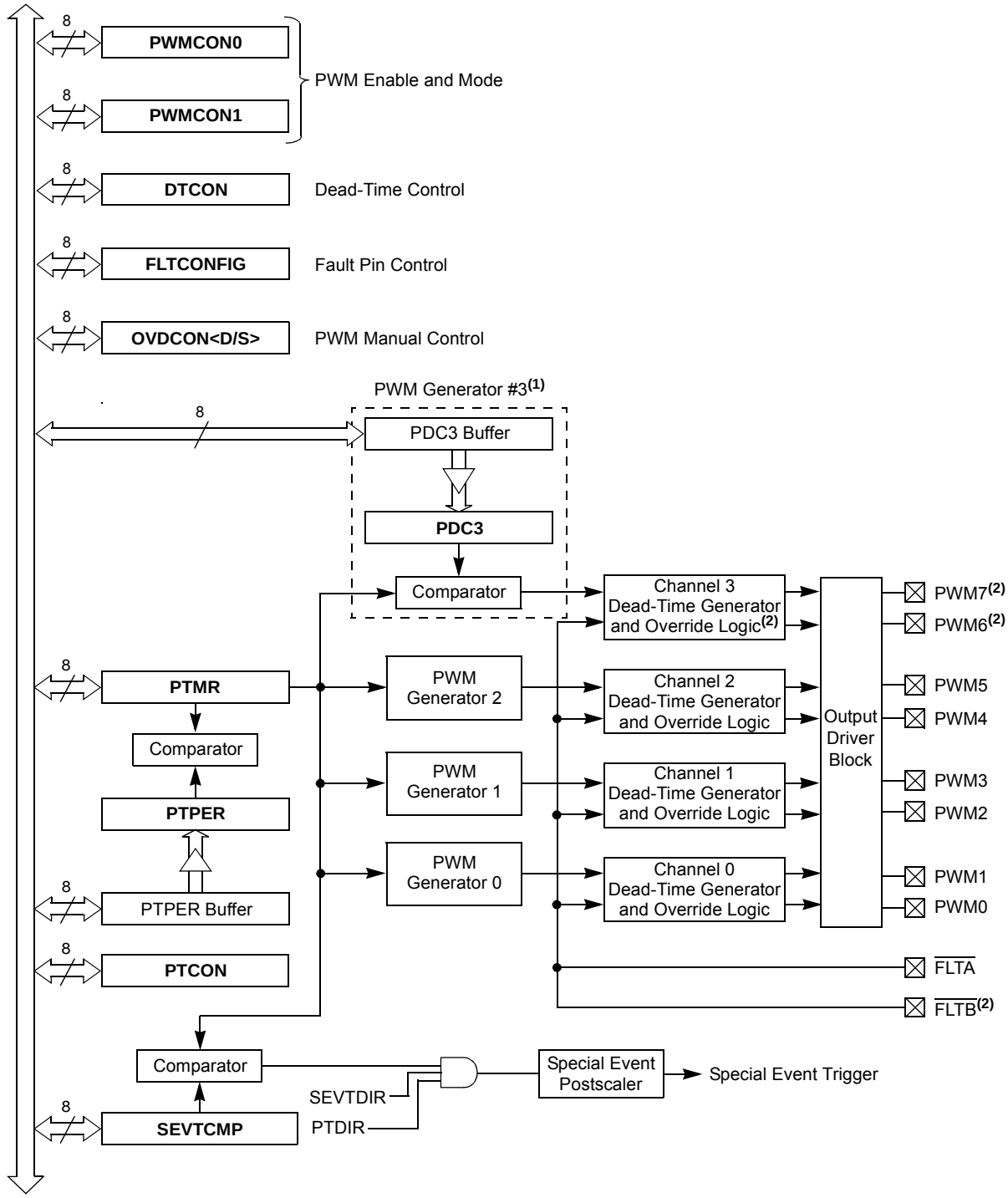
2: RA5 I/F is available only on 40-pin devices (PIC18F4331/4431).

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 11-1: INITIALIZING PORTA

```
CLRF    PORTA   ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF    LATA    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0x3F    ; Configure A/D
MOVWF   ANSEL0  ; for digital inputs
MOVLW   0xCF    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISA   ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
```

FIGURE 18-1: POWER CONTROL PWM MODULE BLOCK DIAGRAM



Note 1: Only PWM Generator 3 is shown in detail. The other generators are identical; their details are omitted for clarity.

2: PWM Generator 3 and its logic, PWM Channels 6 and 7, and $\overline{FLTB}$ and its associated logic are not implemented on PIC18F2331/2431 devices.

18.6 PWM Duty Cycle

PWM duty cycle is defined by the PDCx (PDCxL and PDCxH) registers. There are a total of four PWM Duty Cycle registers for four pairs of PWM channels. The Duty Cycle registers have 14-bit resolution by combining six LSbs of PDCxH with the 8 bits of PDCxL. PDCx is a double-buffered register used to set the counting period for the PWM time base.

18.6.1 PWM DUTY CYCLE REGISTERS

There are four 14-bit Special Function Registers used to specify duty cycle values for the PWM module:

- PDC0 (PDC0L and PDC0H)
- PDC1 (PDC1L and PDC1H)
- PDC2 (PDC2L and PDC2H)
- PDC3 (PDC3L and PDC3H)

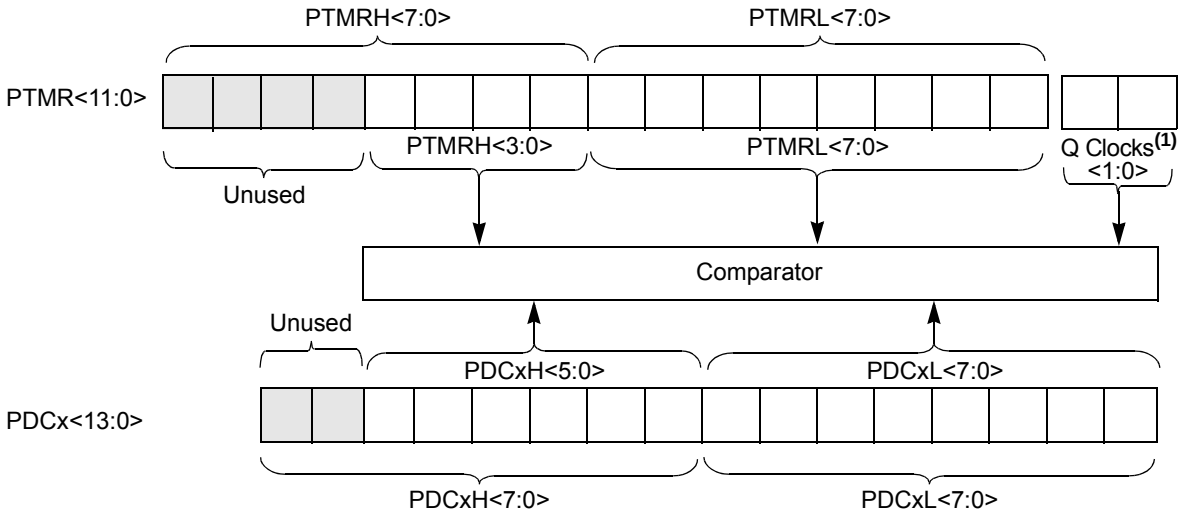
The value in each Duty Cycle register determines the amount of time that the PWM output is in the active state. The upper 12 bits of PDCx holds the actual duty cycle value from PTMRH/L<11:0>, while the lower 2 bits control which internal Q clock the duty cycle match will occur. This 2-bit value is decoded from the Q clocks as shown in Figure 18-11 (when the prescaler is 1:1 or PTCKPS<1:0> = 00).

In Edge-Aligned mode, the PWM period starts at Q1 and ends when the Duty Cycle register matches the PTMR register as follows. The duty cycle match is considered when the upper 12 bits of the PDCx are equal to the PTMR and the lower 2 bits are equal to Q1, Q2, Q3 or Q4, depending on the lower two bits of the PDCx (when the prescaler is 1:1 or PTCKPS<1:0> = 00).

Note:	When the prescaler is not 1:1 (PTCKPS<1:0> ≠ ~00), the duty cycle match occurs at the Q1 clock of the instruction cycle when the PTMR and PDCx match occurs.

Each compare unit has logic that allows override of the PWM signals. This logic also ensures that the PWM signals will complement each other (with dead-time insertion) in Complementary mode (see **Section 18.7 "Dead-Time Generators"**).

FIGURE 18-11: DUTY CYCLE COMPARISON



Note 1: This value is decoded from the Q clocks:
00 = duty cycle match occurs on Q1
01 = duty cycle match occurs on Q2
10 = duty cycle match occurs on Q3
11 = duty cycle match occurs on Q4

18.7 Dead-Time Generators

In power inverter applications, where the PWMs are used in Complementary mode to control the upper and lower switches of a half-bridge, a dead-time insertion is highly recommended. The dead-time insertion keeps both outputs in inactive state for a brief time. This avoids any overlap in the switching during the state change of the power devices due to TON and TOFF characteristics.

Because the power output devices cannot switch instantaneously, some amount of time must be provided between the turn-off event of one PWM output in a complementary pair and the turn-on event of the other transistor. The PWM module allows dead time to be programmed. The following sections explain the dead-time block in detail.

18.7.1 DEAD-TIME INSERTION

Each complementary output pair for the PWM module has a 6-bit down counter used to produce the dead-time insertion. As shown in Figure 18-17, each dead-time unit has a rising and falling edge detector connected to the duty cycle comparison output. The dead time is loaded into the timer on the detected PWM edge event. Depending on whether the edge is rising or falling, one of the transitions on the complementary outputs is delayed until the timer counts down to zero. A timing diagram, indicating the dead-time insertion for one pair of PWM outputs, is shown in Figure 18-18.

FIGURE 18-17: DEAD-TIME CONTROL UNIT BLOCK DIAGRAM FOR ONE PWM OUTPUT PAIR

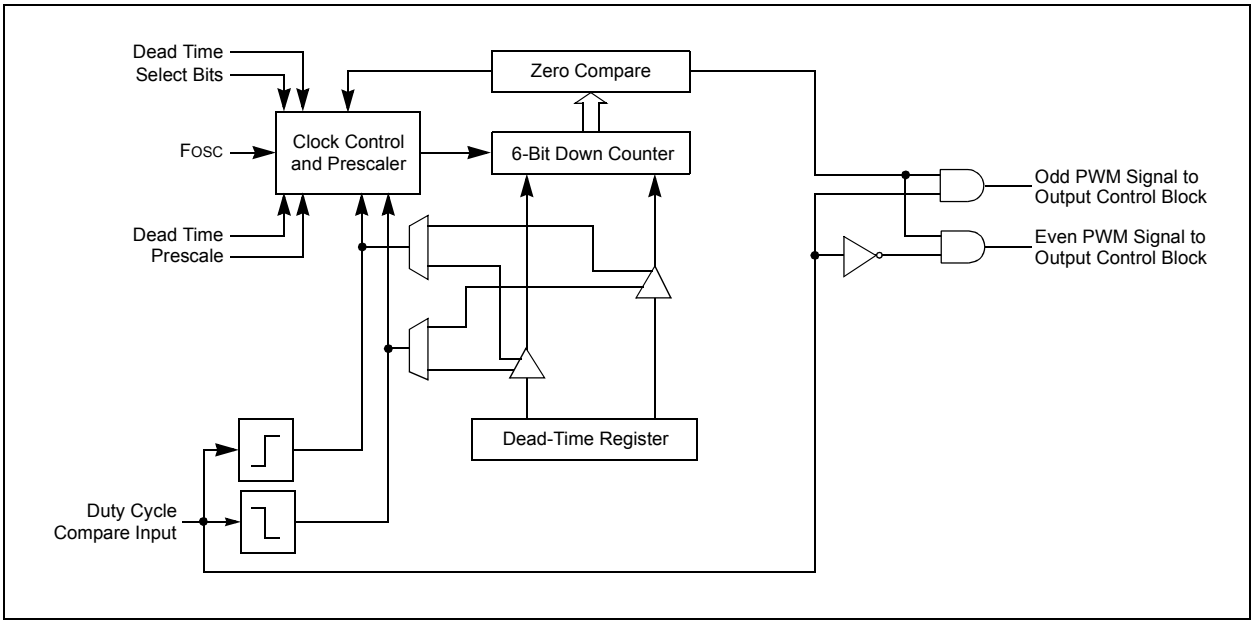
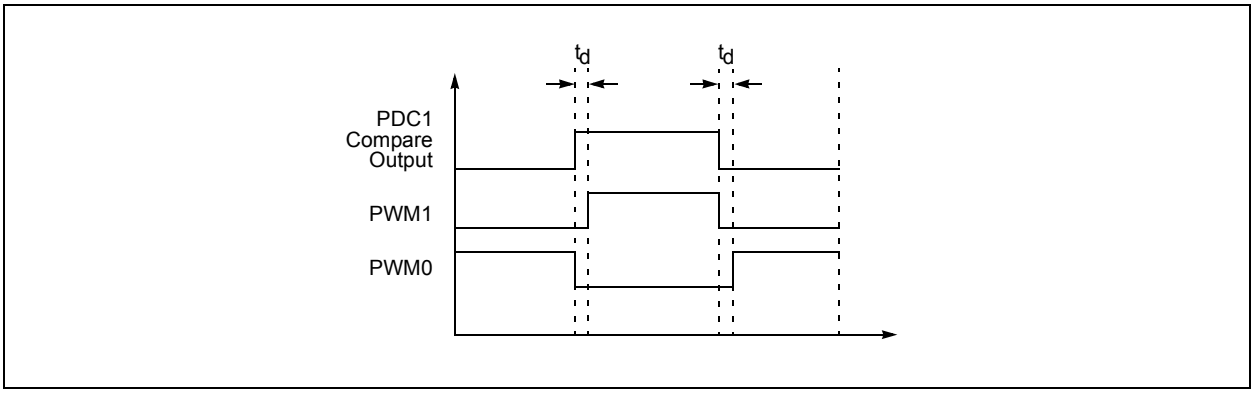


FIGURE 18-18: DEAD-TIME INSERTION FOR COMPLEMENTARY PWM



18.11.3 PWM OUTPUT PIN RESET STATES

The PWMPIN Configuration bit determines the PWM output pins to be PWM output pins or digital I/O pins, after the device comes out of Reset. If the PWMPIN Configuration bit is unprogrammed (default), the PWMEN<2:0> control bits will be cleared on a device Reset. Consequently, all PWM outputs will be tri-stated and controlled by the corresponding PORT and TRIS registers. If the PWMPIN Configuration bit is programmed low, the PWMEN<2:0> control bits will be set, as follows, on a device Reset:

- PWMEN<2:0> = 101 if device has 8 PWM pins (PIC18F4331/4431 devices)
- PWMEN<2:0> = 100 if device has 6 PWM pins (PIC18F2331/2431 devices)

All PWM pins will be enabled for PWM output and will have the output polarity defined by the HPOL and LPOL Configuration bits.

18.12 PWM Fault Inputs

There are two Fault inputs associated with the PWM module. The main purpose of the input Fault pins is to disable the PWM output signals and drive them into an inactive state. The action of the Fault inputs is performed directly in hardware so that when a Fault occurs, it can be managed quickly and the PWM outputs are put into an inactive state to save the power devices connected to the PWMs.

The PWM Fault inputs are $\overline{\text{FLTA}}$ and $\overline{\text{FLTB}}$, which can come from I/O pins, the CPU or another module. The $\overline{\text{FLTA}}$ and $\overline{\text{FLTB}}$ pins are active-low inputs so it is easy to "OR" many sources to the same input. $\overline{\text{FLTB}}$ and its associated logic are not implemented on PIC18F2331/2431 devices.

The FLTCONFIG register (Register 18-8) defines the settings of $\overline{\text{FLTA}}$ and $\overline{\text{FLTB}}$ inputs.

Note:	The inactive state of the PWM pins are dependent on the HPOL and LPOL Configuration bit settings, which define the active and inactive state for PWM outputs.

18.12.1 FAULT PIN ENABLE BITS

By setting the bits, FLTAEN and FLTBEN in the FLTCONFIG register, the corresponding Fault inputs are enabled. If both bits are cleared, then the Fault inputs have no effect on the PWM module.

18.12.2 MFAULT INPUT MODES

The FLTAMOD and FLTBMOD bits in the FLTCONFIG register determine the modes of PWM I/O pins that are deactivated when they are overridden by Fault input.

The FLTAS and FLTBS bits in the FLTCONFIG register give the status of Fault A and Fault B inputs.

Each of the Fault inputs have two modes of operation:

- **Inactive Mode (FLTxMOD = 0)**

  This is a Catastrophic Fault Management mode. When the Fault occurs in this mode, the PWM outputs are deactivated. The PWM pins will remain in Inactivate mode until the Fault is cleared (Fault input is driven high) and the corresponding Fault Status bit has been cleared in software. The PWM outputs are enabled immediately at the beginning of the following PWM period, after the Fault Status bit (FLTxS) is cleared.

- **Cycle-by-Cycle Mode (FLTxMOD = 1)**

  When the Fault occurs in this mode, the PWM outputs are deactivated. The PWM outputs will remain in the defined Fault states (all PWM outputs inactive) for as long as the Fault pin is held low. After the Fault pin is driven high, the PWM outputs will return to normal operation at the beginning of the following PWM period and the FLTxS bit is automatically cleared.

19.3.1.1 Addressing

Once the SSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.
b) The Buffer Full bit, BF, is set.
c) An $\overline{ACK}$ pulse is generated.
d) SSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) on the falling edge of the ninth SCL pulse.

In 10-Bit Addressing mode, two address bytes need to be received by the slave (Figure 19-7). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{W}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address.

The sequence of events for 10-Bit Addressing mode is as follows, with Steps 7-9 for slave-transmitter:

1. Receive first (high) byte of address (SSPIF, BF and UA bits are set).
2. Update the SSPADD register with second (low) byte of address (clears bit, UA, and releases the SCL line).
3. Read the SSPBUF register (clears bit, BF) and clear flag bit, SSPIF.
4. Receive second (low) byte of address (SSPIF, BF and UA bits are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit, UA.
6. Read the SSPBUF register (clears bit, BF) and clear flag bit, SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (SSPIF and BF bits are set).
9. Read the SSPBUF register (clears bit, BF) and clear flag bit, SSPIF.

TABLE 19-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set SSPIF Bit (SSP interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

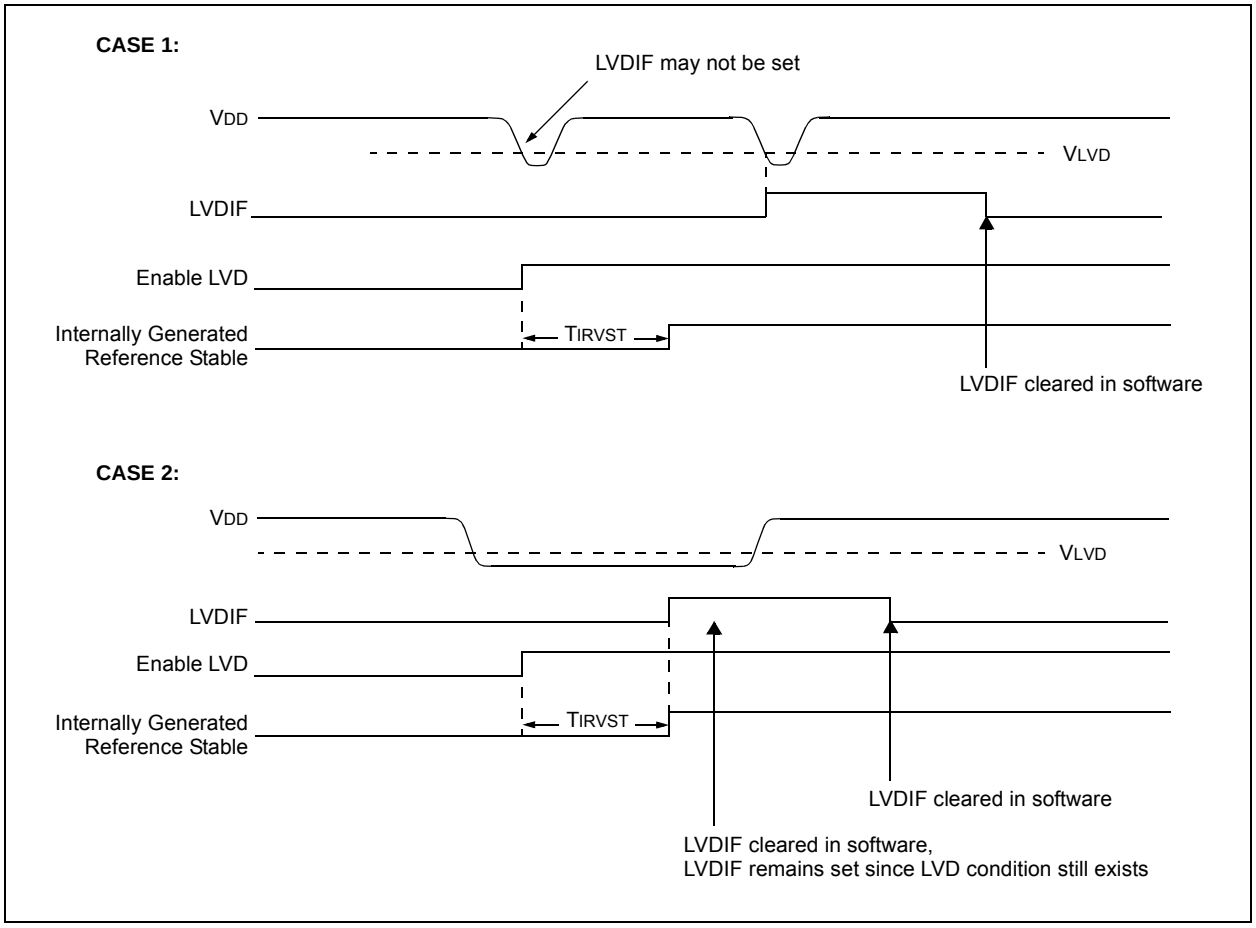
Note: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

22.4 LVD Start-up Time

The internal reference voltage of the LVD module, specified in electrical specification Parameter D420, may be used by other internal circuitry, such as the Programmable Brown-out Reset. If the LVD, or other circuits using the voltage reference, are disabled to lower the device's current consumption, the reference voltage circuit will require time to become stable before a low-voltage condition can be reliably detected. This start-up time, TIRVST, is an interval that is independent of device clock speed. It is specified in electrical specification Parameter 36.

The LVD interrupt flag is not enabled until TIRVST has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval (refer to Figure 22-2).

FIGURE 22-2: LOW-VOLTAGE DETECT WAVEFORMS



REGISTER 23-6: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVREN
bit 7							bit 0

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

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

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

bit 2 **LVP:** Single-Supply ICSP™ Enable bit
1 = Single-Supply ICSP is enabled
0 = Single-Supply ICSP is disabled

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

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

REGISTER 23-11: CONFIG7L: CONFIGURATION REGISTER 7 LOW (BYTE ADDRESS 30000Ch)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	EBTR3(1,2,3)	EBTR2(1,2,3)	EBTR1(2,3)	EBTR0(2,3)
bit 7							bit 0

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

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

bit 3 **EBTR3:** Table Read Protection bit(1,2,3)
1 = Block 3 is not protected from table reads executed in other blocks
0 = Block 3 is protected from table reads executed in other blocks

bit 2 **EBTR2:** Table Read Protection bit(1,2,3)
1 = Block 2 is not protected from table reads executed in other blocks
0 = Block 2 is protected from table reads executed in other blocks

bit 1 **EBTR1:** Table Read Protection bit(2,3)
1 = Block 1 is not protected from table reads executed in other blocks
0 = Block 1 is protected from table reads executed in other blocks

bit 0 **EBTR0:** Table Read Protection bit(2,3)
1 = Block 0 is not protected from table reads executed in other blocks
0 = Block 0 is protected from table reads executed in other blocks

Note 1: Unimplemented in PIC18F2331/4331 devices; maintain this bit set.
2: Refer to Figure 23-5 for block boundary addresses.
3: Enabling the corresponding CPx bit is recommended to protect the block from external read operations.

REGISTER 23-12: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)

U-0	R/P-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB(1,2)	—	—	—	—	—	—
bit 7							bit 0

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

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

bit 6 **EBTRB:** Boot Block Table Read Protection bit(1,2)
1 = Boot block is not protected from table reads executed in other blocks
0 = Boot block is protected from table reads executed in other blocks

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

Note 1: Enabling the corresponding CPx bit is recommended to protect the block from external read operations.
2: Refer to Figure 23-5 for block boundary addresses.

TABLE 24-2: PIC18FXXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb LSb	Status Affected	Notes
LITERAL OPERATIONS						
ADDLW	k	Add Literal and WREG	1	0000 1111 kkkk kkkk	C, DC, Z, OV, N	
ANDLW	k	AND Literal with WREG	1	0000 1011 kkkk kkkk	Z, N	
IORLW	k	Inclusive OR Literal with WREG	1	0000 1001 kkkk kkkk	Z, N	
LFSR	f, k	Load Literal (12-bit) 2nd word to FSRx 1st word	2	1110 1110 00ff kkkk 1111 0000 kkkk kkkk	None	
MOVLB	k	Move Literal to BSR<3:0>	1	0000 0001 0000 kkkk	None	
MOVLW	k	Move Literal to WREG	1	0000 1110 kkkk kkkk	None	
MULLW	k	Multiply Literal with WREG	1	0000 1101 kkkk kkkk	None	
RETLW	k	Return with Literal in WREG	2	0000 1100 kkkk kkkk	None	
SUBLW	k	Subtract WREG from Literal	1	0000 1000 kkkk kkkk	C, DC, Z, OV, N	
XORLW	k	Exclusive OR Literal with WREG	1	0000 1010 kkkk kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS						
TBLRD*		Table Read	2	0000 0000 0000 1000	None	
TBLRD*+		Table Read with Post-Increment		0000 0000 0000 1001	None	
TBLRD*-		Table Read with Post-Decrement		0000 0000 0000 1010	None	
TBLRD+*		Table Read with Pre-Increment		0000 0000 0000 1011	None	
TBLWT*		Table Write	2 (5)	0000 0000 0000 1100	None	
TBLWT*+		Table Write with Post-Increment		0000 0000 0000 1101	None	
TBLWT*-		Table Write with Post-Decrement		0000 0000 0000 1110	None	
TBLWT+*		Table Write with Pre-Increment		0000 0000 0000 1111	None	

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as an input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

4: Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

24.2 Instruction Set

ADDLW **ADD Literal to W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

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

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

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal ‘k’ and 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: ADDLW 0x15

Before Instruction
W = 0x10
After Instruction
W = 0x25

ADDWF **ADD W to f**

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

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

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

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

Encoding:

0010	01da	ffff	ffff

Description: Add W to register, ‘f’. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register, ‘f’. If ‘a’ is ‘0’, the Access Bank will be selected. If ‘a’ is ‘1’, the BSR is used.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: ADDWF REG, W

Before Instruction
W = 0x17
REG = 0xC2
After Instruction
W = 0xD9
REG = 0xC2

TBLWT Table Write

Syntax: [*label*] TBLWT (*; *+; *-; +*)

Operands: None

Operation: if TBLWT*,
(TABLAT) → Holding Register,
TBLPTR – No Change;
if TBLWT*+,
(TABLAT) → Holding Register,
(TBLPTR) + 1 → TBLPTR;
if TBLWT*-,
(TABLAT) → Holding Register,
(TBLPTR) – 1 → TBLPTR;
if TBLWT+*,
(TBLPTR) + 1 → TBLPTR,
(TABLAT) → Holding Register

Status Affected: None

Encoding:

0000	0000	0000	11nn nn = 0 * =1 *+ =2 *- =3 +*

Description: This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 8.0 "Flash Program Memory"** for additional details on programming Flash memory.)
The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLWT instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

TBLWT Table Write (Continued)

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

Example 1: TBLWT *+;

Before Instruction
TABLAT = 0x55
TBLPTR = 0x00A356
HOLDING REGISTER (0x00A356) = 0xFF

After Instructions (table write completion)
TABLAT = 0x55
TBLPTR = 0x00A357
HOLDING REGISTER (0x00A356) = 0x55

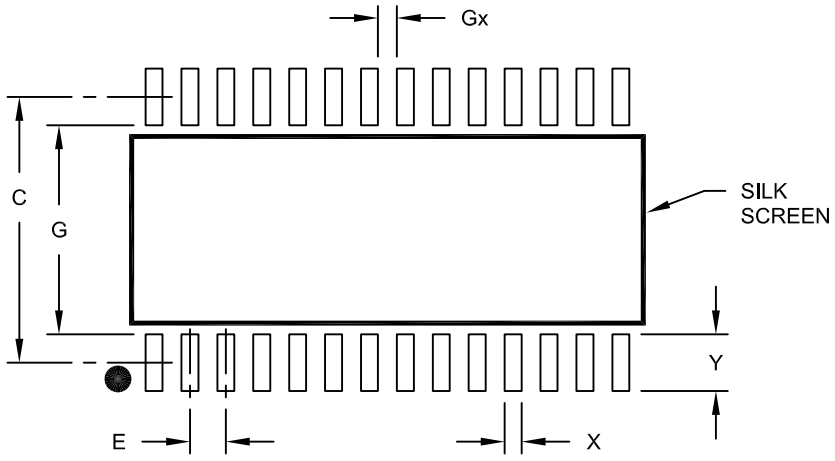
Example 2: TBLWT +*;

Before Instruction
TABLAT = 0x34
TBLPTR = 0x01389A
HOLDING REGISTER (0x01389A) = 0xFF
HOLDING REGISTER (0x01389B) = 0xFF

After Instruction (table write completion)
TABLAT = 0x34
TBLPTR = 0x01389B
HOLDING REGISTER (0x01389A) = 0xFF
HOLDING REGISTER (0x01389B) = 0x34

28-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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com