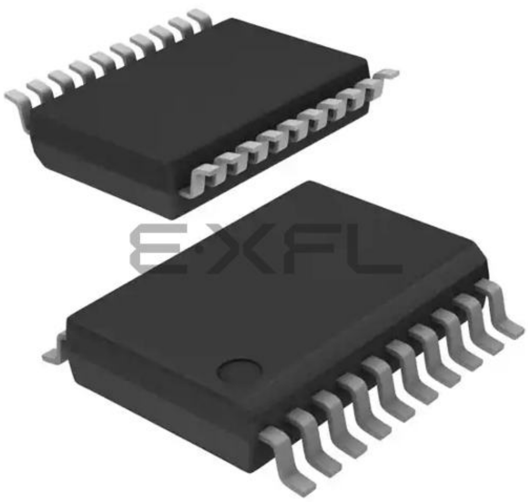


**Welcome to E-XFL.COM**

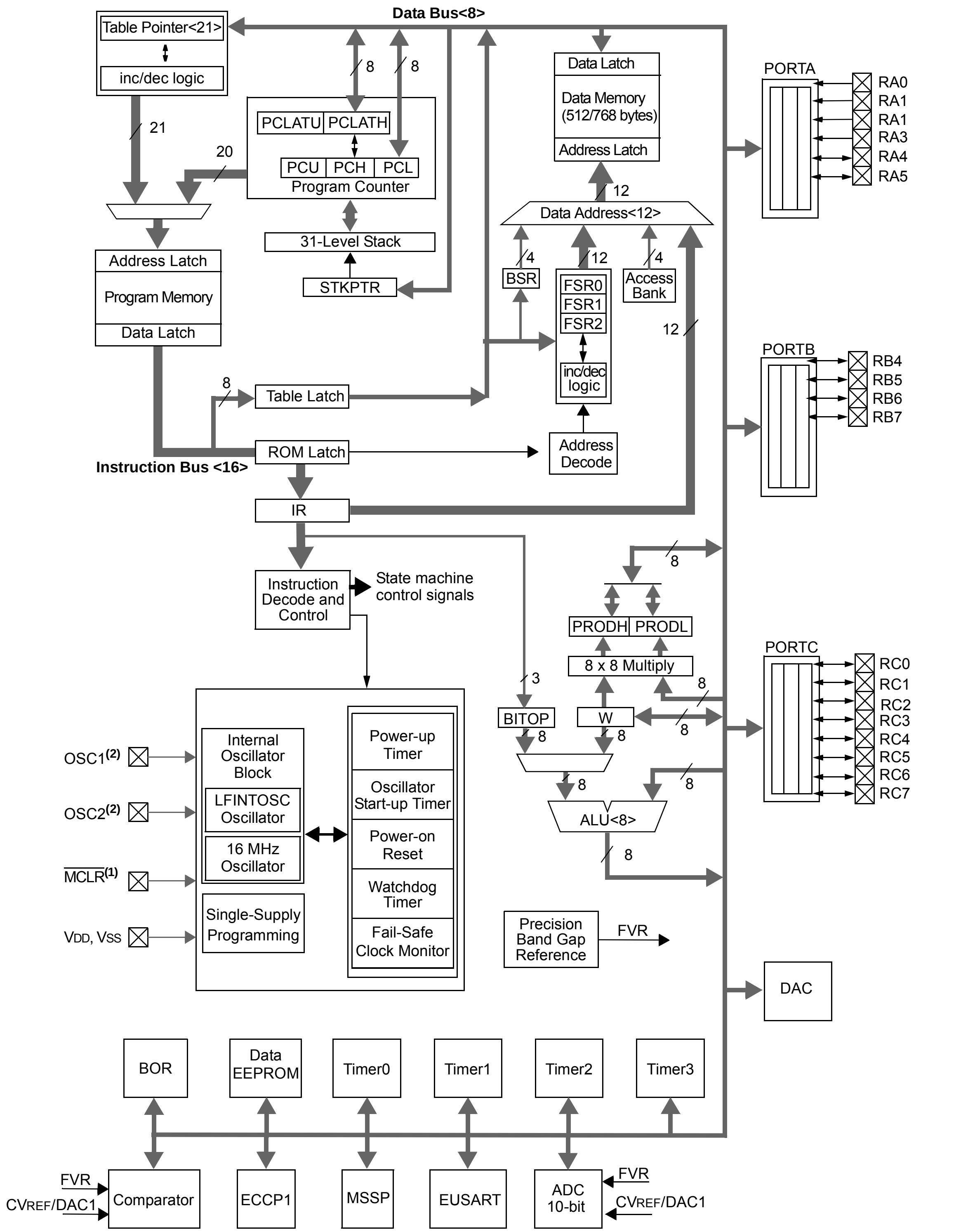
### What is "Embedded - Microcontrollers"?

"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 | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 17 |
| Program Memory Size | 8KB (4K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 20-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f13k22t-i-ss |

**FIGURE 1-1: PIC18(L)F1XK22 BLOCK DIAGRAM**




**Note 1:** RA3 is only available when $\overline{\text{MCLR}}$ functionality is disabled.

**2:** OSC1/CLKIN and OSC2/CLKOUT are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 2.0 "Oscillator Module"** for additional information.

## 3.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 Enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 4.0 "Flash Program Memory"**. Data EEPROM is discussed separately in **Section 5.0 "Data EEPROM Memory"**.

## 3.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte Program Memory (PC) space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

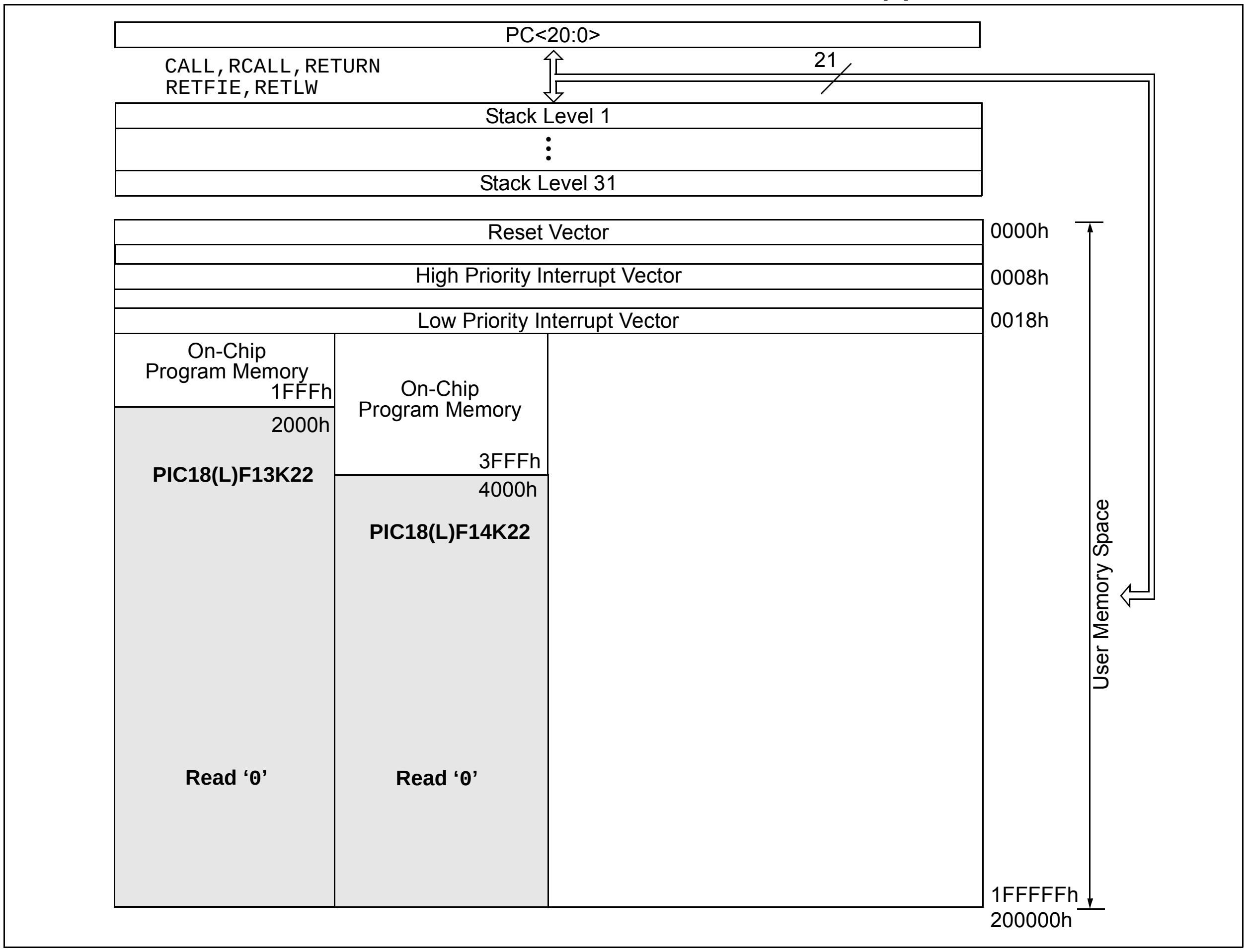
This family of devices contain the following:

- PIC18(L)F13K22: 8 Kbytes of Flash Memory, up to 4,096 single-word instructions
- PIC18(L)F14K22: 16 Kbytes of Flash Memory, up to 8,192 single-word instructions

PIC18 devices have two interrupt vectors and one Reset vector. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory map for PIC18(L)F1XK22 devices is shown in Figure 3-1. Memory block details are shown in Figure 3-2.

**FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC18(L)F1XK22 DEVICES**



# 4.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed one byte at a time. A write to program memory is executed on blocks of 16 or 8 bytes at a time depending on the specific device (See Table 4-1). Program memory is erased in blocks of 64 bytes at a time. The difference between the write and erase block sizes requires from 4 to 8 block writes to restore the contents of a single block erase. A Bulk Erase operation can not be issued from user code.

**TABLE 4-1: WRITE/ERASE BLOCK SIZES**

| Device | Write Block Size (bytes) | Erase Block Size (bytes) |
|---|---|---|
| PIC18(L)F13K22 | 8 | 64 |
| PIC18(L)F14K22 | 16 | 64 |

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

## 4.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

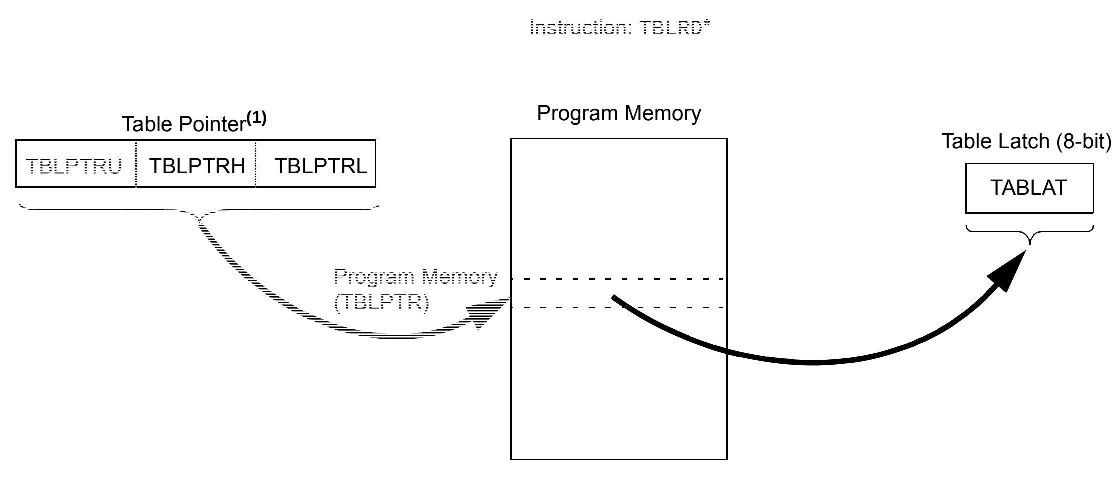
The program memory space is 16-bit wide, while the data RAM space is 8-bit wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 4-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 4.5 "Writing to Flash Program Memory"**. Figure 4-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word-aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned.

**FIGURE 4-1: TABLE READ OPERATION**




Note 1: Table Pointer register points to a byte in program memory.

**TABLE 8-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---|---|---|---|---|---|---|---|---|---|
| ANSEL | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 247 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RABIE | TMR0IF | INT0IF | RABIF | 244 |
| INTCON2 | $\overline{\text{RABPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | — | TMR0IP | — | RABIP | 244 |
| IOCA | — | — | IOCA5 | IOCA4 | IOCA3(2) | IOCA2 | IOCA1 | IOCA0 | 247 |
| LATA | — | — | LATA5(1) | LATA4(1) | — | LATA2 | LATA1 | LATA0 | 247 |
| PORTA | — | — | RA5(1) | RA4(1) | RA3(2) | RA2 | RA1 | RA0 | 247 |
| SLRCON | — | — | — | — | — | SLRC | SLRB | SLRA | 247 |
| TRISA | — | — | TRISA5(1) | TRISA4(1) | —(3) | TRISA2 | TRISA1 | TRISA0 | 247 |
| WPUA | — | — | WPUA5 | WPUA4 | WPUA3(2) | WPUA2 | WPUA1 | WPUA0 | 244 |

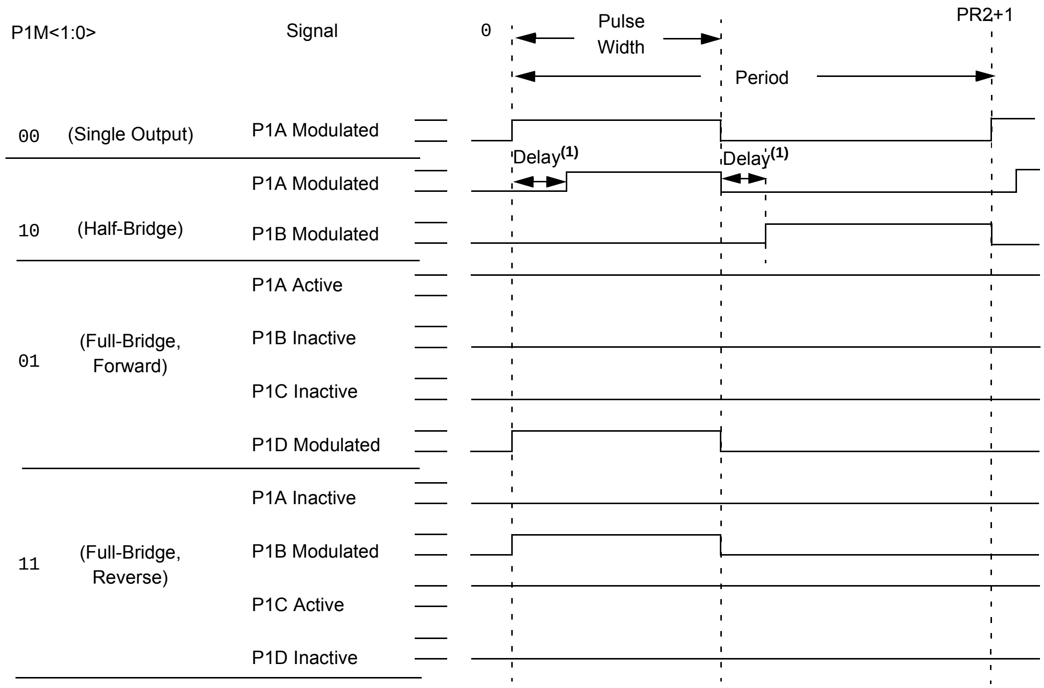
**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTA.

**Note 1:** RA<5:4> and their associated latch and data direction bits are enabled as I/O pins based on oscillator configuration; otherwise, they are read as '0'.

**2:** Implemented only when Master Clear functionality is disabled (MCLRE Configuration bit = 0).

**3:** Unimplemented, read as '1'.

**FIGURE 13-4: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)**




Relationships:

- Period = 4 * TOSC * (PR2 + 1) * (TMR2 Prescale Value)
- Pulse Width = TOSC * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * TOSC * (PWM1CON<6:0>)

**Note 1:** Dead-band delay is programmed using the PWM1CON register (**Section 13.4.6 "Programmable Dead-Band Delay Mode"**).

### 13.4.7 PULSE STEERING MODE

In Single Output mode, pulse steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can be simultaneously available on multiple pins.

Once the Single Output mode is selected (CCP1M<3:2> = 11 and P1M<1:0> = 00 of the CCP1CON register), the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STR<D:A> bits of the PSTRCON register, as shown in Table 13-2.

> **Note:** The associated TRIS bits must be set to output ('0') to enable the pin output driver in order to see the PWM signal on the pin.

While the PWM Steering mode is active, CCP1M<1:0> bits of the CCP1CON register select the PWM output polarity for the P1<D:A> pins.

The PWM auto-shutdown operation also applies to PWM Steering mode as described in **Section 13.4.4 "Enhanced PWM Auto-shutdown mode"**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

**REGISTER 13-4: PSTRCON: PULSE STEERING CONTROL REGISTER(1)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | STRSYNC | STRD | STRC | STRB | STRA |
| bit 7 | | | | | | | bit 0 |

| 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 |

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

bit 4 **STRSYNC:** Steering Sync bit
1 = Output steering update occurs on next PWM period
0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STRD:** Steering Enable bit D
1 = P1D pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1D pin is assigned to port pin

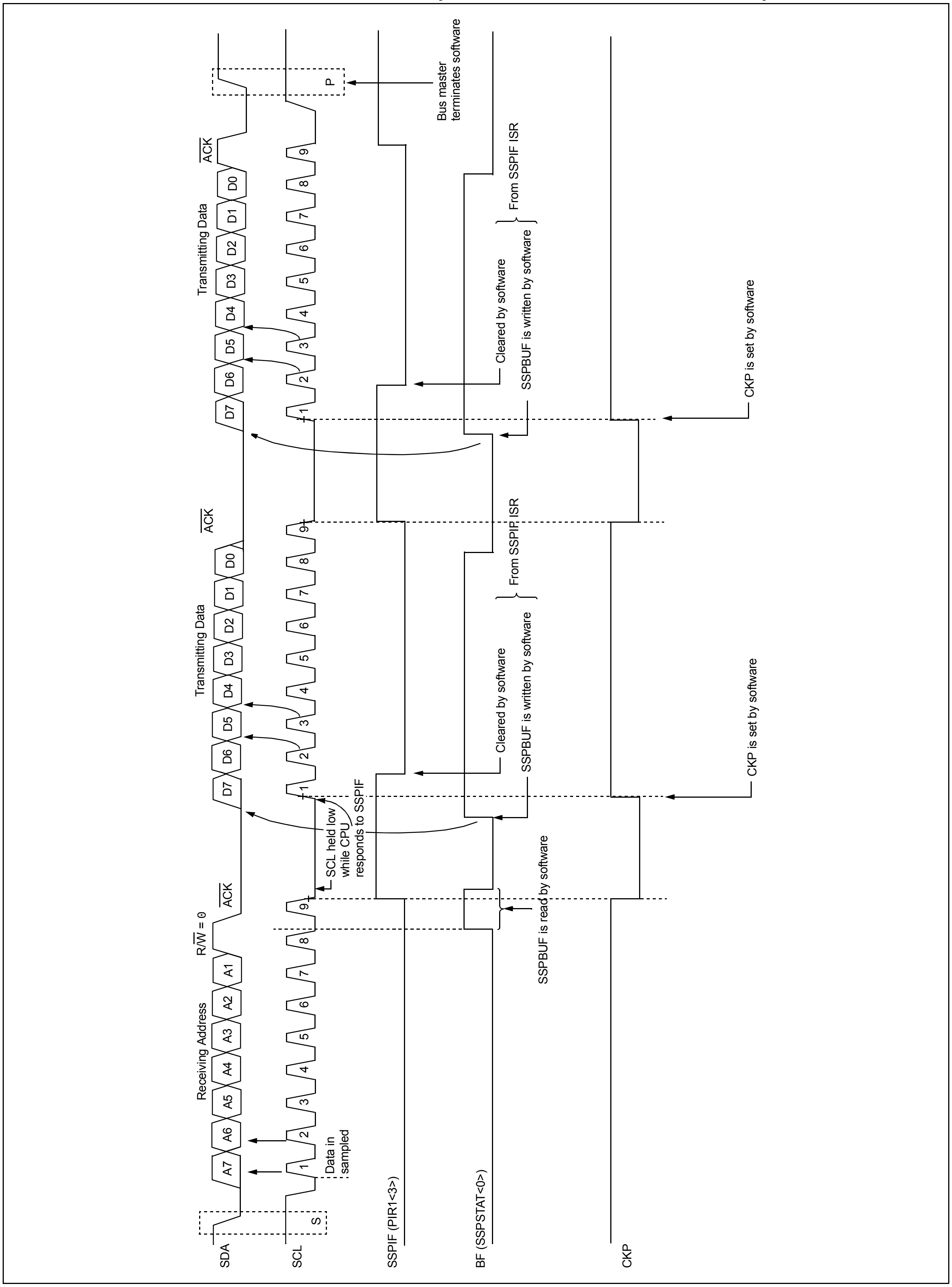
bit 2 **STRC:** Steering Enable bit C
1 = P1C pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1C pin is assigned to port pin

bit 1 **STRB:** Steering Enable bit B
1 = P1B pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1B pin is assigned to port pin

bit 0 **STRA:** Steering Enable bit A
1 = P1A pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1A pin is assigned to port pin

**Note 1:** The PWM Steering mode is available only when the CCP1CON register bits CCP1M<3:2> = 11 and P1M<1:0> = 00.

**FIGURE 14-9: $I^2C$ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)**



**REGISTER 14-7: SSPADD: MSSP ADDRESS AND BAUD RATE REGISTER ($I^2C$ MODE)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADD7 | ADD6 | ADD5 | ADD4 | ADD3 | ADD2 | ADD1 | ADD0 |
| bit 7 | | | | | | | bit 0 |

| **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 |

**Master mode:**

bit 7-0 **ADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) *4)/FOSC

**10-Bit Slave mode — Most Significant Address Byte:**

bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a “don’t care.” Bit pattern sent by master is fixed by $I^2C$ specification and must be equal to ‘11110’. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **ADD<9:8>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care.”

**10-Bit Slave mode — Least Significant Address Byte:**

bit 7-0 **ADD<7:0>:** Eight Least Significant bits of 10-bit address

**7-Bit Slave mode:**

bit 7-1 **ADD<6:0>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care.”

**REGISTER 15-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| bit 7 | | | | | | | bit 0 |

| **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 |

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

**FIGURE 21-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM**

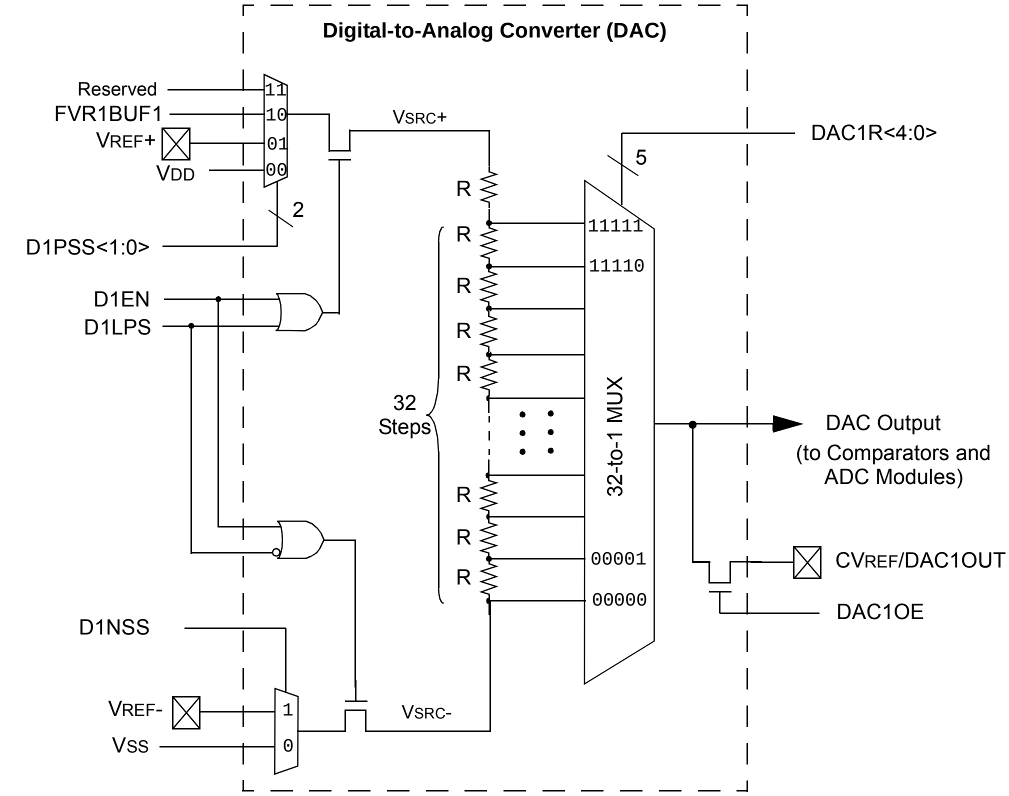



**FIGURE 21-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE**

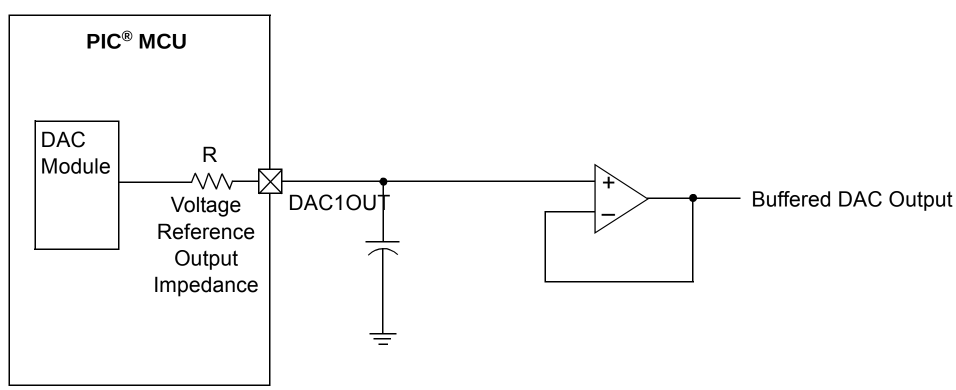


**FIGURE 22-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD, VDD RISE < TPWRT)**

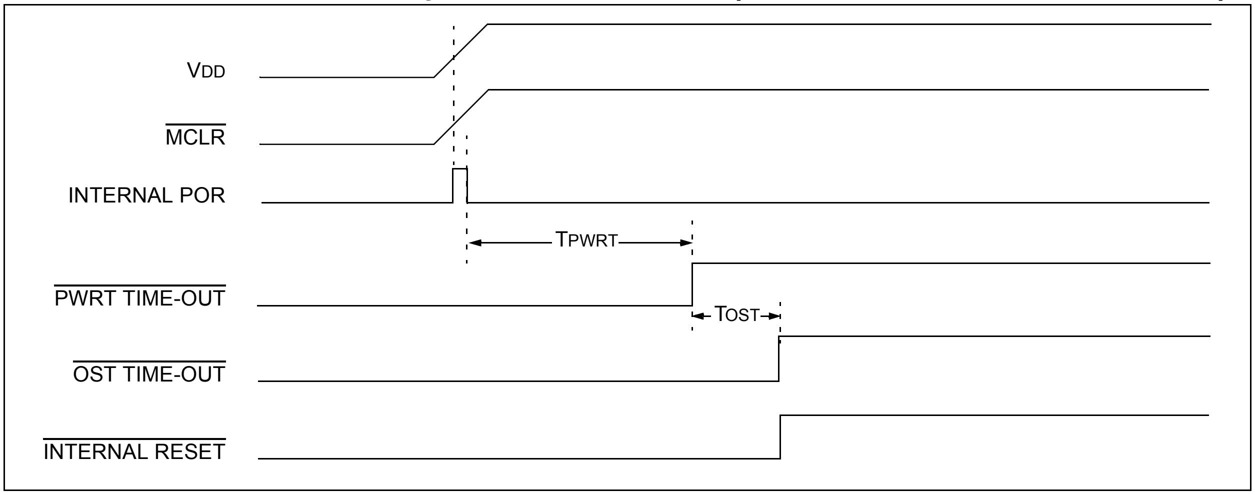



**FIGURE 22-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 1**

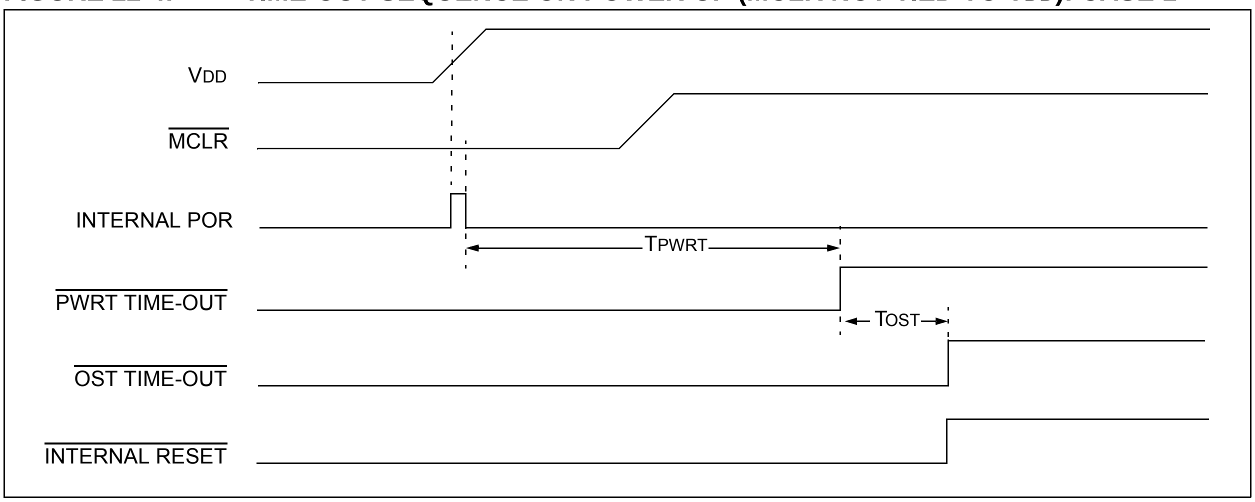
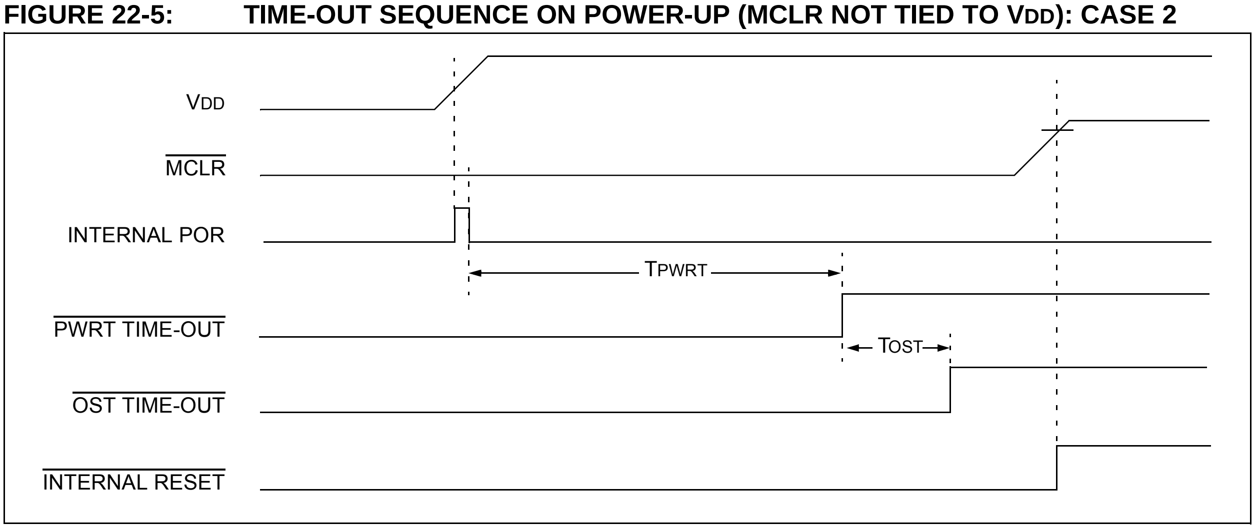



**FIGURE 22-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 2**



## 23.0 SPECIAL FEATURES OF THE CPU

PIC18(L)F1XK22 devices include several features intended to maximize reliability and minimize cost through elimination of external components. These are:

- Oscillator Selection
- Resets:
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Code Protection
- ID Locations
- In-Circuit Serial Programming™

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 2.0 "Oscillator Module"**.

A complete discussion of device Resets and interrupts is available in previous sections of this data sheet.

In addition to their Power-up and Oscillator Start-up Timers provided for Resets, PIC18(L)F1XK22 devices have a Watchdog Timer, which is either permanently enabled via the Configuration bits or software controlled (if configured as disabled).

The inclusion of an internal RC oscillator also provides the additional benefits of a Fail-Safe Clock Monitor (FSCM) and Two-Speed Start-up. FSCM provides for background monitoring of the peripheral clock and automatic switchover in the event of its failure. Two-Speed Start-up enables code to be executed almost immediately on start-up, while the primary clock source completes its start-up delays.

All of these features are enabled and configured by setting the appropriate Configuration register bits.

## BNOV — Branch if Not Overflow

Syntax: BNOV n

Operands: $-128 \leq n \leq 127$

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

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 1110 | 0101 | nnnn | nnnn |

Description: If the OVERFLOW bit is '0', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

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

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'n' | Process Data | No operation |

Example: `HERE BNOV Jump`

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

## BNZ — Branch if Not Zero

Syntax: BNZ n

Operands: $-128 \leq n \leq 127$

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

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 1110 | 0001 | nnnn | nnnn |

Description: If the ZERO bit is '0', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

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

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'n' | Process Data | No operation |

Example: `HERE BNZ Jump`

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

## SUBWFB — Subtract W from f with Borrow

Syntax: SUBWFB f {,d {,a}}

Operands: $0 \leq f \leq 255$
d ∈ [0,1]
a ∈ [0,1]

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

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

Encoding:

| | | | |
|---|---|---|---|
| 0101 | 10da | ffff | ffff |

Description: Subtract W and the CARRY flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: `SUBWFB REG, 1, 0`

Before Instruction
REG = 19h (0001 1001)
W = 0Dh (0000 1101)
C = 1
After Instruction
REG = 0Ch (0000 1011)
W = 0Dh (0000 1101)
C = 1
Z = 0
N = 0 ; result is positive

Example 2: `SUBWFB REG, 0, 0`

Before Instruction
REG = 1Bh (0001 1011)
W = 1Ah (0001 1010)
C = 0
After Instruction
REG = 1Bh (0001 1011)
W = 00h
C = 1
Z = 1 ; result is zero
N = 0

Example 3: `SUBWFB REG, 1, 0`

Before Instruction
REG = 03h (0000 0011)
W = 0Eh (0000 1101)
C = 1
After Instruction
REG = F5h (1111 0100) ; [2's comp]
W = 0Eh (0000 1101)
C = 0
Z = 0
N = 1 ; result is negative

## SWAPF — Swap f

Syntax: SWAPF f {,d {,a}}

Operands: $0 \leq f \leq 255$
d ∈ [0,1]
a ∈ [0,1]

Operation: (f<3:0>) → dest<7:4>,
(f<7:4>) → dest<3:0>

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0011 | 10da | ffff | ffff |

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `SWAPF REG, 1, 0`

Before Instruction
REG = 53h
After Instruction
REG = 35h

| XORWF | Exclusive OR W with f |
|---|---|
| Syntax: | XORWF f {,d {,a}} |
| Operands: | $0 \leq f \leq 255$<br>d ∈ [0,1]<br>a ∈ [0,1] |
| Operation: | (W) .XOR. (f) → dest |
| Status Affected: | N, Z |
| Encoding: | `0001` `10da` `ffff` `ffff` |
| Description: | Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).<br>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).<br>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details. |
| Words: | 1 |
| Cycles: | 1 |

Q Cycle Activity:

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

Example: `XORWF REG, 1, 0`

Before Instruction
- REG = AFh
- W = B5h

After Instruction
- REG = 1Ah
- W = B5h

### 24.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set of the PIC18(L)F1XK22 family of devices. This includes the MPLAB® C18 C compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing mode. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option, or dialog box within the environment, that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.

### TABLE 26-9: I/O PORTS

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Sym.** | **Characteristic** | **Min.** | **Typ.†** | **Max.** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O ports: | | | | | |
| D036 | | with TTL buffer | VSS | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D036A | | | VSS | — | 0.15 VDD | V | 1.8V ≤ VDD ≤ 4.5V |
| D036B | | | VSS | — | 0.2 VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| D037 | | with Schmitt Trigger buffer | VSS | — | 0.2 VDD | V | 1.8V ≤ VDD ≤ 5.5V |
| D037A | | with $I^2C$ levels | VSS | — | 0.3 VDD | V | |
| D037B | | with SMBus levels | VSS | — | 0.8 VDD | V | 2.7V ≤ VDD ≤ 5.5V |
| D038 | | $\overline{MCLR}$ | VSS | — | 0.2 VDD | V | |
| D039 | | OSC1 | VSS | — | 0.3 VDD | V | HS, HSPLL modes |
| D039A | | OSC1 | VSS | — | 0.2 VDD | V | EC, RC modes(1) |
| D039B | | OSC1 | VSS | — | 0.3 VDD | V | XT, LP modes |
| D039C | | T1CKI | VSS | — | 0.3 VDD | V | |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O ports: | | | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8 | — | VDD | V | 1.8V ≤ VDD ≤ 4.5V |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | V | 1.8V ≤ VDD ≤ 5.5V |
| D041A | | with $I^2C$ levels | 0.7 VDD | — | VDD | V | |
| D037A | | with SMBus levels | 2.1 | — | VDD | V | 2.7V ≤ VDD ≤ 5.5V |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | VDD | V | |
| D042A | | $\overline{MCLR}$ | 0.9 VDD | — | 0.3 VDD | V | 1.8V ≤ VDD ≤ 2.4V |
| D043 | | OSC1 | 0.7 VDD | — | VDD | V | HS, HSPLL modes |
| D043A | | OSC1 | 0.8 VDD | — | VDD | V | EC mode |
| D043B | | OSC1 | 0.9 VDD | — | VDD | V | RC mode(1) |
| D043C | | OSC1 | 1.6 | — | VDD | V | XT, LP modes |
| D043E | | T1CKI | 1.6 | — | VDD | V | |
| | IIL | **Input Leakage Current(2)** | | | | | |
| D060 | | I/O ports | — | ± 5 | ± 100 | nA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, -40°C to 85°C |
| | | | — | ± 5 | ± 1000 | nA | VSS ≤ VPIN ≤ VDD, 85°C to 125°C |
| D061 | | $\overline{MCLR}$(3) | — | ± 50 | ± 200 | nA | VSS ≤ VPIN ≤ VDD |
| | IPUR | **PORTB Weak Pull-up Current** | | | | | |
| D070* | | | 50 | 250 | 400 | μA | VDD = 5.0V, VPIN = VSS |

* These parameters are characterized but not tested.

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

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

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

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

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

**FIGURE 26-9: CLKOUT AND I/O TIMING**

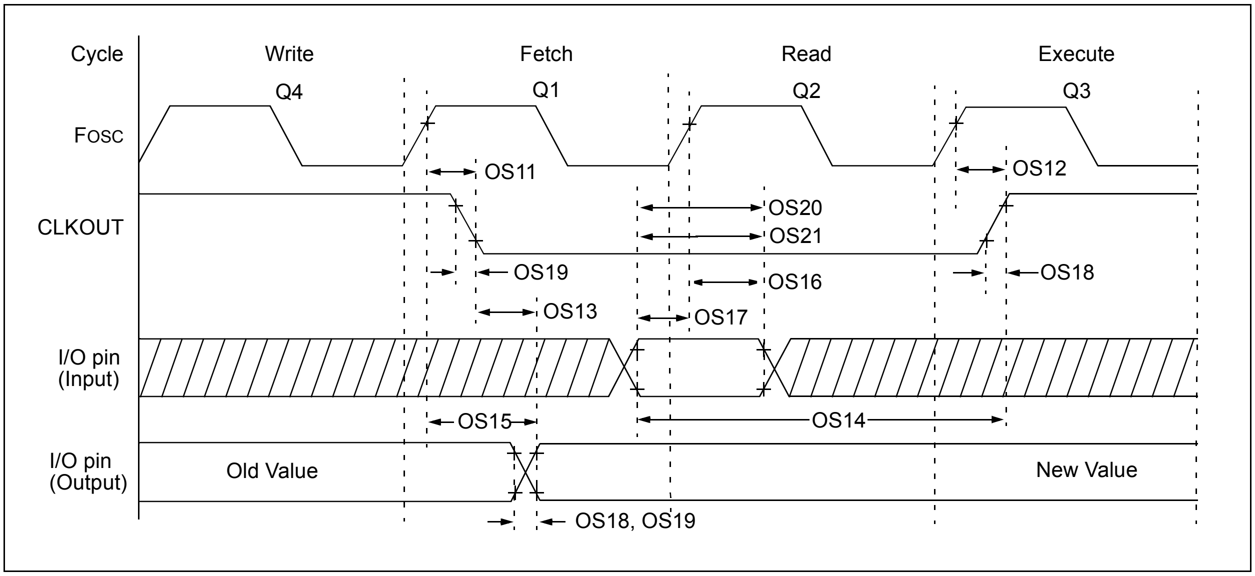



**TABLE 26-15: CLKOUT AND I/O TIMING PARAMETERS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Sym. | Characteristic | Min. | Typ.† | Max. | Units | Conditions |
| OS11 | TOSH2CKL | Fosc↑ to CLKOUT↓ (1) | — | — | 70 | ns | VDD = 3.3-5.0V |
| OS12 | TOSH2CKH | Fosc↑ to CLKOUT↑ (1) | — | — | 72 | ns | VDD = 3.3-5.0V |
| OS13 | TCKL2IOV | CLKOUT↓ to Port out valid(1) | — | — | 20 | ns | |
| OS14 | TIOV2CKH | Port input valid before CLKOUT↑(1) | TOSC + 200 ns | — | — | ns | |
| OS15 | TOSH2IOV | Fosc↑ (Q1 cycle) to Port out valid | — | 50 | 70* | ns | VDD = 3.3-5.0V |
| OS16 | TOSH2IOI | Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 50 | — | — | ns | VDD = 3.3-5.0V |
| OS17 | TIOV2OSH | Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time) | 20 | — | — | ns | |
| OS18 | TIOR | Port output rise time(2) | —<br>— | 40<br>15 | 72<br>32 | ns | VDD = 1.8V<br>VDD = 3.3-5.0V |
| OS19 | TIOF | Port output fall time(2) | —<br>— | 28<br>15 | 55<br>30 | ns | VDD = 1.8V<br>VDD = 3.3-5.0V |
| OS20* | TINP | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | TRBP | PORTB interrupt-on-change new input level time | 25 | — | — | ns | |

* These parameters are characterized but not tested.
† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated.
**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.
**2:** Includes OSC2 in CLKOUT mode.

**FIGURE 27-5: PIC18LF1XK22 ICOMP – TYPICAL IPD FOR COMPARATOR IN LOW-POWER MODE**

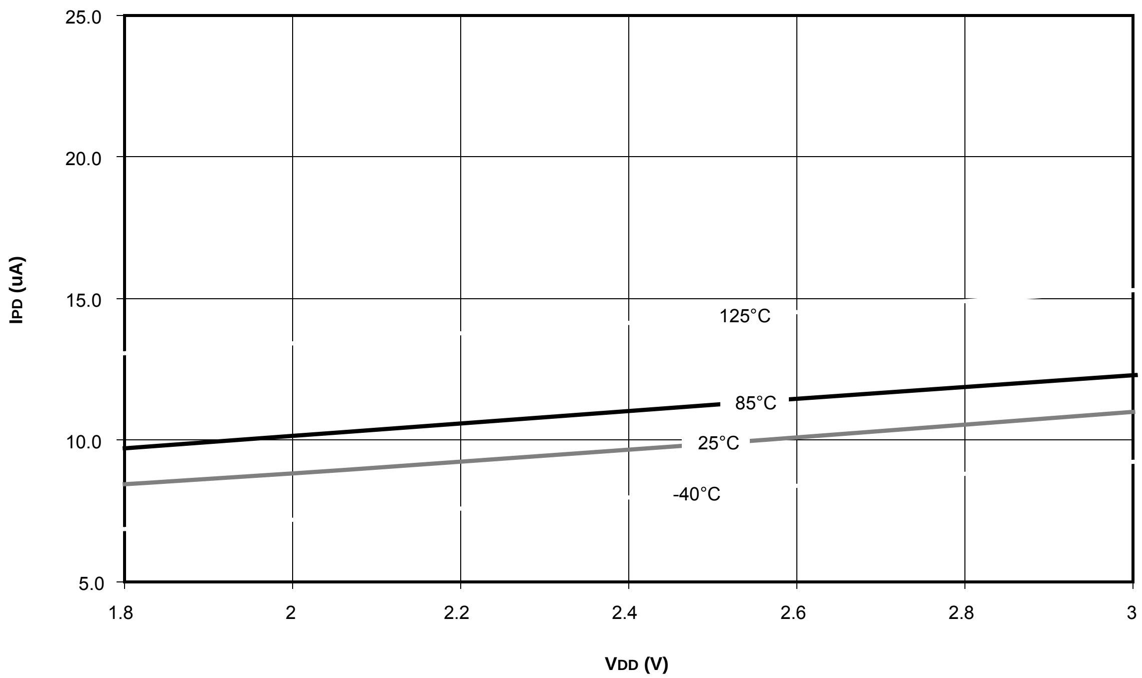
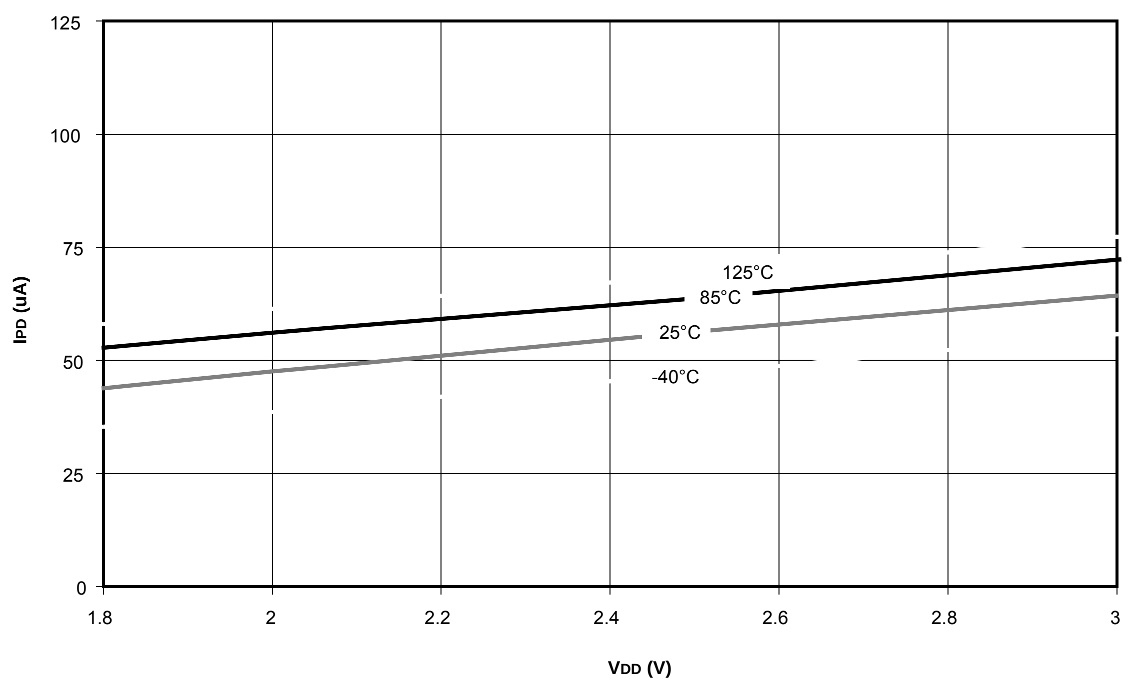
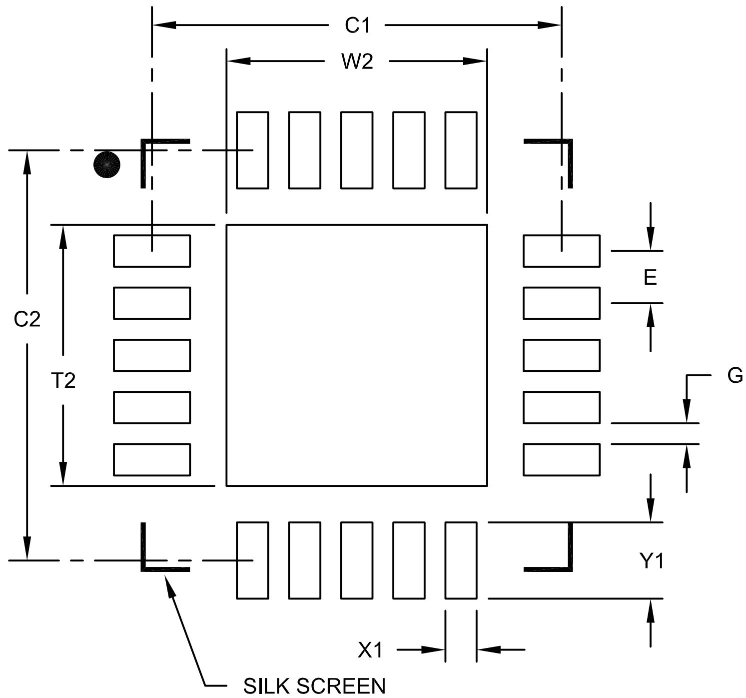



**FIGURE 27-6: PIC18LF1XK22 ICOMP – TYPICAL IPD FOR COMPARATOR IN HIGH-POWER MODE**



20-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4 mm Body [QFN]
With 0.40 mm Contact Length

| **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 | 0.50 BSC | | |
| Optional Center Pad Width | W2 | | | 2.50 |
| Optional Center Pad Length | T2 | | | 2.50 |
| Contact Pad Spacing | C1 | | 3.93 | |
| Contact Pad Spacing | C2 | | 3.93 | |
| Contact Pad Width | X1 | | | 0.30 |
| Contact Pad Length | Y1 | | | 0.73 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2126A