



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

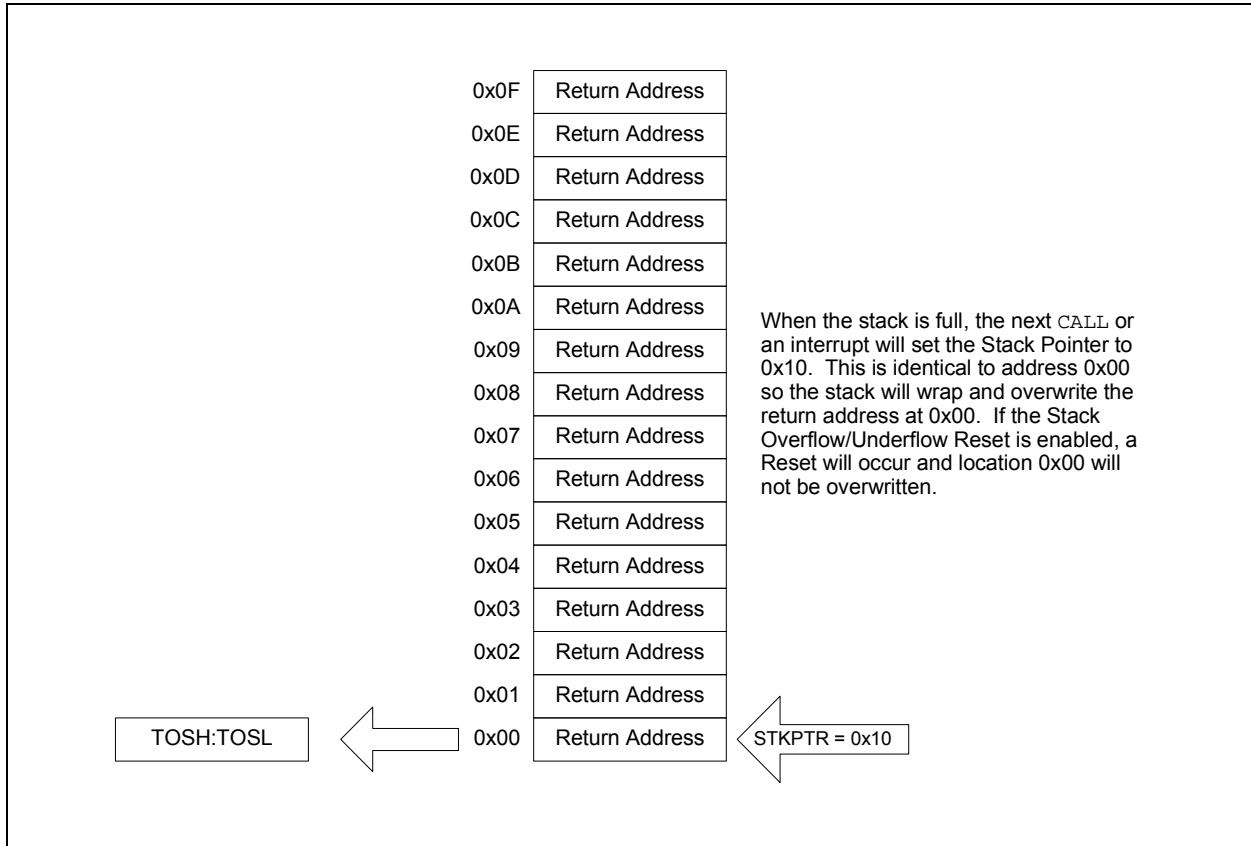
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1829-e-ss

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



3.4.2 OVERFLOW/UNDERFLOW RESET

If the `STVREN` bit in Configuration Word 2 is programmed to '1', the device will be reset if the stack is `PUSHed` beyond the sixteenth level or `POPed` beyond the first level, setting the appropriate bits (`STKOVF` or `STKUNF`, respectively) in the `PCON` register.

3.5 Indirect Addressing

The `INDFn` registers are not physical registers. Any instruction that accesses an `INDFn` register actually accesses the register at the address specified by the File Select Registers (`FSR`). If the `FSRn` address specifies one of the two `INDFn` registers, the read will return '0' and the write will not occur (though Status bits may be affected). The `FSRn` register value is created by the pair `FSRnH` and `FSRnL`.

The `FSR` registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

7.3 MCLR

The $\overline{\text{MCLR}}$ is an optional external input that can reset the device. The $\overline{\text{MCLR}}$ function is controlled by the MCLRE bit of Configuration Word 1 and the LVP bit of Configuration Word 2 (Table 7-2).

TABLE 7-2: MCLR CONFIGURATION

MCLRE	LVP	MCLR
0	0	Disabled
1	0	Enabled
x	1	Enabled

7.3.1 MCLR ENABLED

When $\overline{\text{MCLR}}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{\text{MCLR}}$ pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

Note: A Reset does not drive the $\overline{\text{MCLR}}$ pin low.

7.3.2 MCLR DISABLED

When $\overline{\text{MCLR}}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 12.2 “PORTA Registers”** for more information.

7.4 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDTC instruction within the time-out period. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register are changed to indicate the WDT Reset. See **Section 10.0 “Watchdog Timer”** for more information.

7.5 RESET Instruction

A RESET instruction will cause a device Reset. The $\overline{\text{RI}}$ bit in the PCON register will be set to ‘0’. See Table 7-4 for default conditions after a RESET instruction has occurred.

7.6 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Word 2. See **Section 3.4.2 “Overflow/Underflow Reset”** for more information.

7.7 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

7.8 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overline{\text{PWRTS}}$ bit of Configuration Word 1.

7.9 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. $\overline{\text{MCLR}}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 5.0 “Oscillator Module (With Fail-Safe Clock Monitor)”** for more information.

The Power-up Timer and oscillator start-up timer run independently of $\overline{\text{MCLR}}$ Reset. If $\overline{\text{MCLR}}$ is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing $\overline{\text{MCLR}}$ high, the device will begin execution immediately (see Figure 7-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

Even if the flag bits were checked before executing a **SLEEP** instruction, it may be possible for flag bits to become set before the **SLEEP** instruction completes. To determine whether a **SLEEP** instruction executed, test the **PD** bit. If the **PD** bit is set, the **SLEEP** instruction was executed as a **NOP**.

FIGURE 9-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT

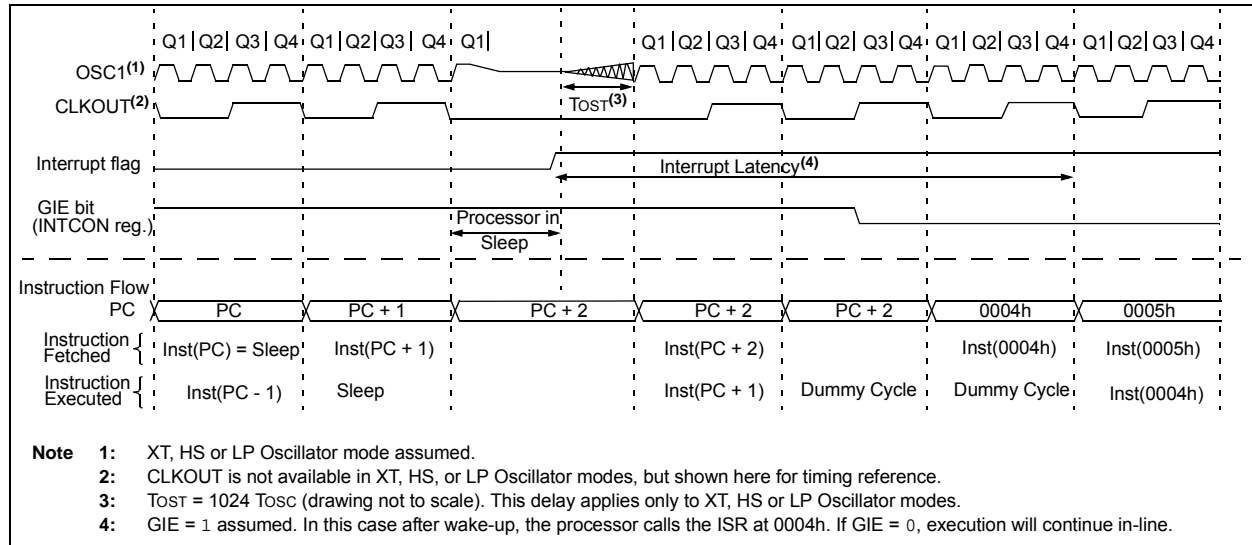


TABLE 9-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCF	87
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	138
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	138
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	137
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	139
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	139
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	138
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	88
PIE2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	CCP2IE	89
PIE4 ⁽¹⁾	—	—	—	—	—	—	BCL2IE	SSP2IE	91
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	92
PIR2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	CCP2IF	93
PIR4 ⁽¹⁾	—	—	—	—	—	—	BCL2IF	SSP2IF	95
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	22
WDTCON	—	—	WDTPS<4:0>					SWDTEN	100

Legend: — Unimplemented, read as '0'. Shaded cells are not used in Power-Down mode.

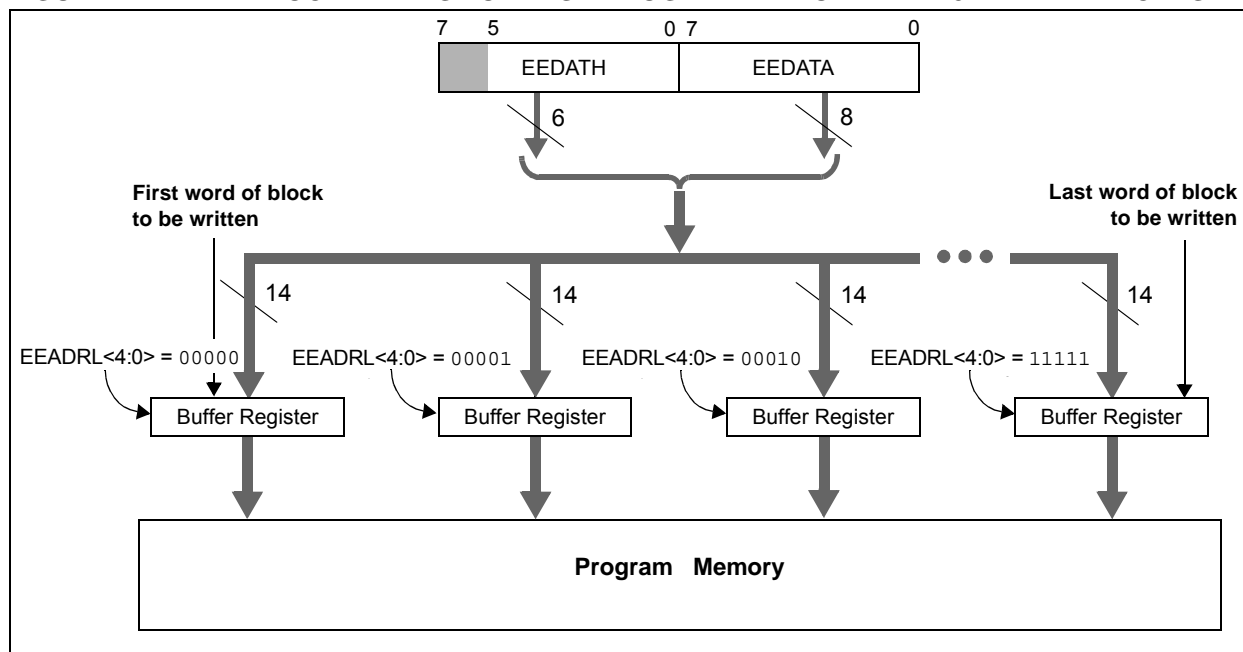
Note 1: PIC16(L)F1829 only.

PIC16(L)F1825/9

After the "BSF EECON1,WR" instruction, the processor requires two cycles to set up the write operation. The user must place two NOP instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms, only during the cycle in which the write takes place (i.e., the last word of the block write). This is not Sleep mode as the clocks and peripherals will

continue to run. The processor does not stall when LWLO = 1, loading the write latches. After the write cycle, the processor will resume operation with the third instruction after the EECON1 write instruction.

FIGURE 11-2: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 32 WRITE LATCHES



11.7 EEPROM and Flash Control Registers

REGISTER 11-1: EEDATL: EEPROM LOW BYTE DATA REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
EEDAT<7:0>							
bit 7							bit 0

Legend:

R = Readable bit
 u = Bit is unchanged
 '1' = Bit is set
 W = Writable bit
 x = Bit is unknown
 '0' = Bit is cleared
 U = Unimplemented bit, read as '0'
 -n/n = Value at POR and BOR/Value at all other Resets

bit 7-0 **EEDAT<7:0>**: Read/write value for EEPROM data byte or Least Significant bits of program memory

REGISTER 11-2: EEDATH: EEPROM DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	EEDAT<13:8>					
bit 7							bit 0

Legend:

R = Readable bit
 u = Bit is unchanged
 '1' = Bit is set
 W = Writable bit
 x = Bit is unknown
 '0' = Bit is cleared
 U = Unimplemented bit, read as '0'
 -n/n = Value at POR and BOR/Value at all other Resets

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

bit 5-0 **EEDAT<13:8>**: Read/write value for Most Significant bits of program memory

REGISTER 11-3: EEADRL: EEPROM ADDRESS REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EEADR<7:0>							
bit 7							bit 0

Legend:

R = Readable bit
 u = Bit is unchanged
 '1' = Bit is set
 W = Writable bit
 x = Bit is unknown
 '0' = Bit is cleared
 U = Unimplemented bit, read as '0'
 -n/n = Value at POR and BOR/Value at all other Resets

bit 7-0 **EEADR<7:0>**: Specifies the Least Significant bits for program memory address or EEPROM address

REGISTER 11-4: EEADRH: EEPROM ADDRESS HIGH BYTE REGISTER

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—(1)	EEADR<14:8>						
bit 7							bit 0

Legend:

R = Readable bit
 u = Bit is unchanged
 '1' = Bit is set
 W = Writable bit
 x = Bit is unknown
 '0' = Bit is cleared
 U = Unimplemented bit, read as '0'
 -n/n = Value at POR and BOR/Value at all other Resets

bit 7 **Unimplemented**: Read as '1'

bit 6-0 **EEADR<14:8>**: Specifies the Most Significant bits for program memory address or EEPROM address

Note 1: Unimplemented, read as '1'.

PIC16(L)F1825/9

12.0 I/O PORTS

Depending on the device selected and peripherals enabled, there are up to two ports available. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

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

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)
- INLVx (input level control)

TABLE 12-1: PORT AVAILABILITY PER DEVICE

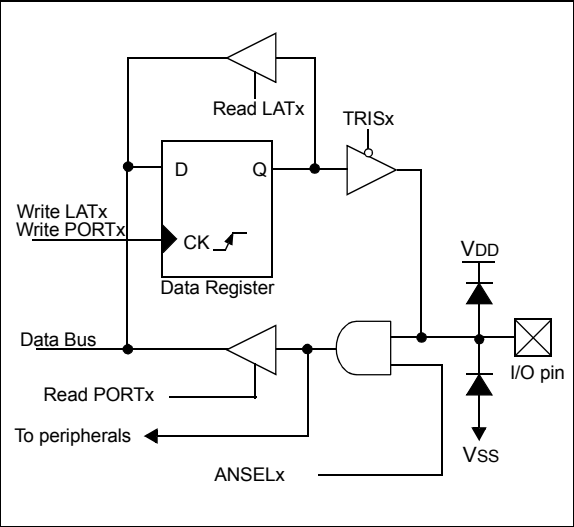
Device	PORTA	PORTB	PORTC
PIC16(L)F1825	•		•
PIC16(L)F1829	•	•	•

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

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 12-1.

FIGURE 12-1: GENERIC I/O PORT OPERATION



EXAMPLE 12-1: INITIALIZING PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL PORTA      ;
CLRF PORTA         ;Init PORTA
BANKSEL LATA        ;Data Latch
CLRF LATA          ;
BANKSEL ANSELA      ;
CLRF ANSELA        ;digital I/O
BANKSEL TRISA       ;
MOVLW B'00111000'  ;Set RA<5:3> as inputs
MOVWF TRISA        ;and set RA<2:0> as
                  ;outputs
```

PIC16(L)F1825/9

12.4.2 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the priority list.

TABLE 12-7: PORTC OUTPUT PRIORITY

Pin Name	Function Priority ⁽¹⁾
RC0	SCL (PIC16(L)F1825 only) SCK (PIC16(L)F1825 only) P1D ⁽²⁾
RC1	SDA1 (PIC16(L)F1825 only) SDA1 (PIC16(L)F1825 only) P1C ⁽²⁾
RC2	SDO1 ⁽²⁾ (PIC16(L)F1825 only) P1D ⁽²⁾ P2B ⁽²⁾
RC3	SS1 ⁽²⁾ (PIC16(L)F1825 only) CCP2 ⁽²⁾ P1C ⁽²⁾ P2A ⁽²⁾
RC4	MDOUT SRNQ C2OUT TX ⁽²⁾ /CK ⁽²⁾ P1B
RC5	RX ⁽²⁾ /DT ⁽²⁾ CCP1/P1A
RC6 ⁽³⁾	SS1 CCP4
RC7 ⁽³⁾	SDO1

- Note**
- 1: Priority listed from highest to lowest.
 - 2: Pin function is selectable via the APFCON0 or APFCON1 register.
 - 3: PIC16(L)F1829 only.

PIC16(L)F1825/9

REGISTER 16-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-0 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

REGISTER 16-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>		—	—	—	—	—	—
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

21.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

21.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

Note: The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to using Timer1. A suitable delay similar to the OST delay can be implemented in software by clearing the TMR1IF bit then presetting the TMR1H:TMR1L register pair to FC00h. The TMR1IF flag will be set when 1024 clock cycles have elapsed, thereby indicating that the oscillator is running and reasonably stable.

21.5 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 21.5.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”**).

Note: When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

21.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

21.6 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 gate can also be driven by multiple selectable sources.

21.6.1 TIMER1 GATE ENABLE

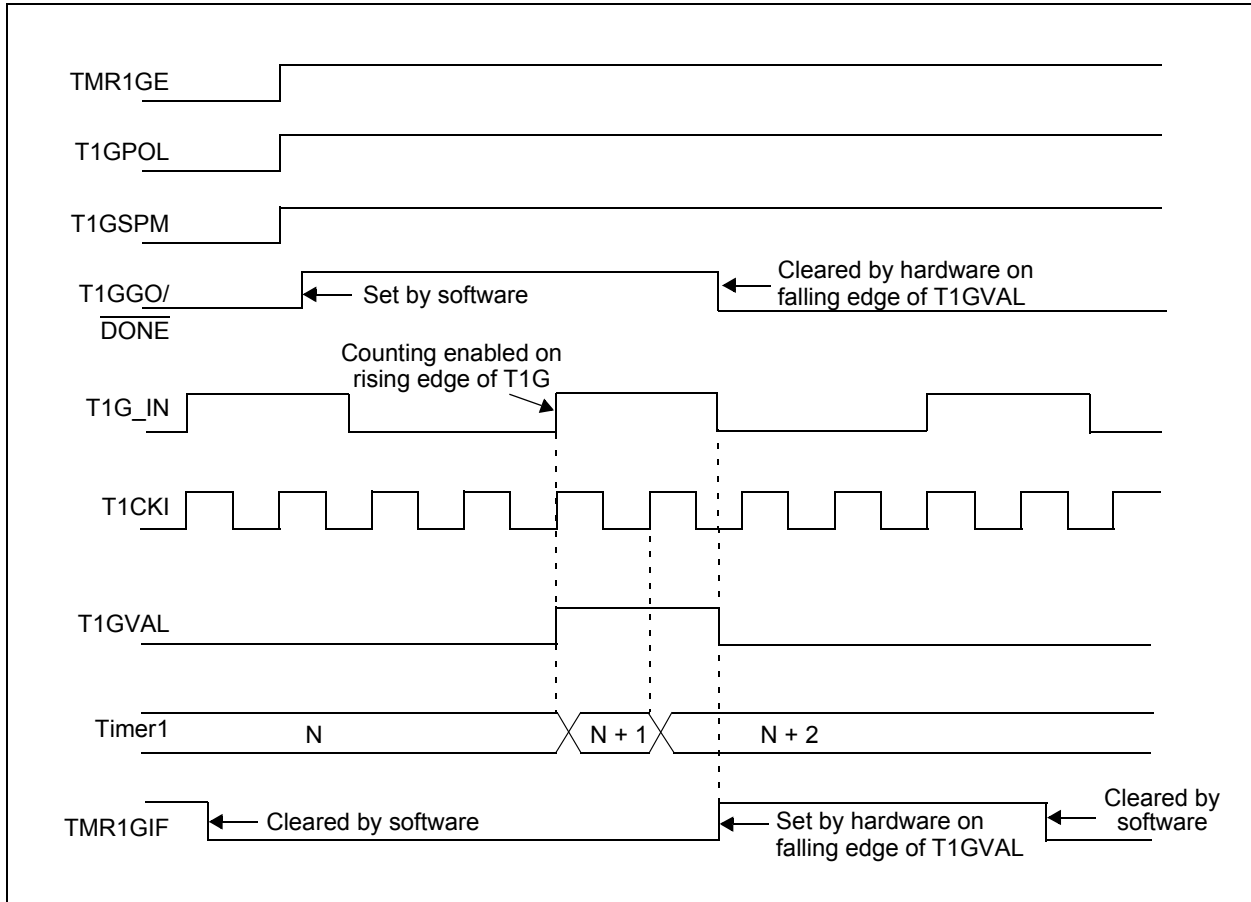
The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 21-3 for timing details.

TABLE 21-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

FIGURE 21-5: TIMER1 GATE SINGLE-PULSE MODE



24.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCPx/PxA pin, while the complementary PWM output signal is output on the PxB pin (see Figure 24-9). This mode can be used for half-bridge applications, as shown in Figure 24-9, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of the PDC<6:0> bits of the PWMxCON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 24.4.5 “Programmable Dead-Band Delay Mode”** for more details of the dead-band delay operations.

Since the PxA and PxB outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure PxA and PxB as outputs.

FIGURE 24-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT

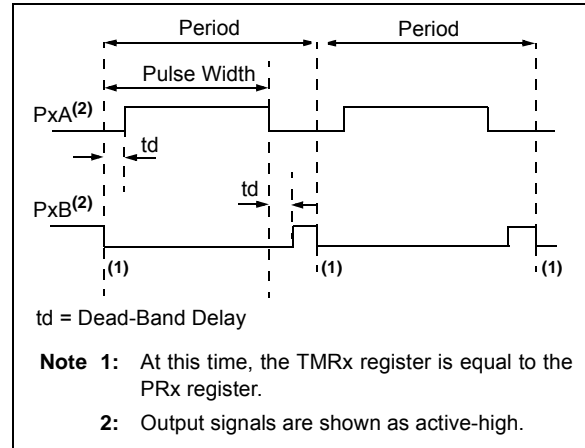
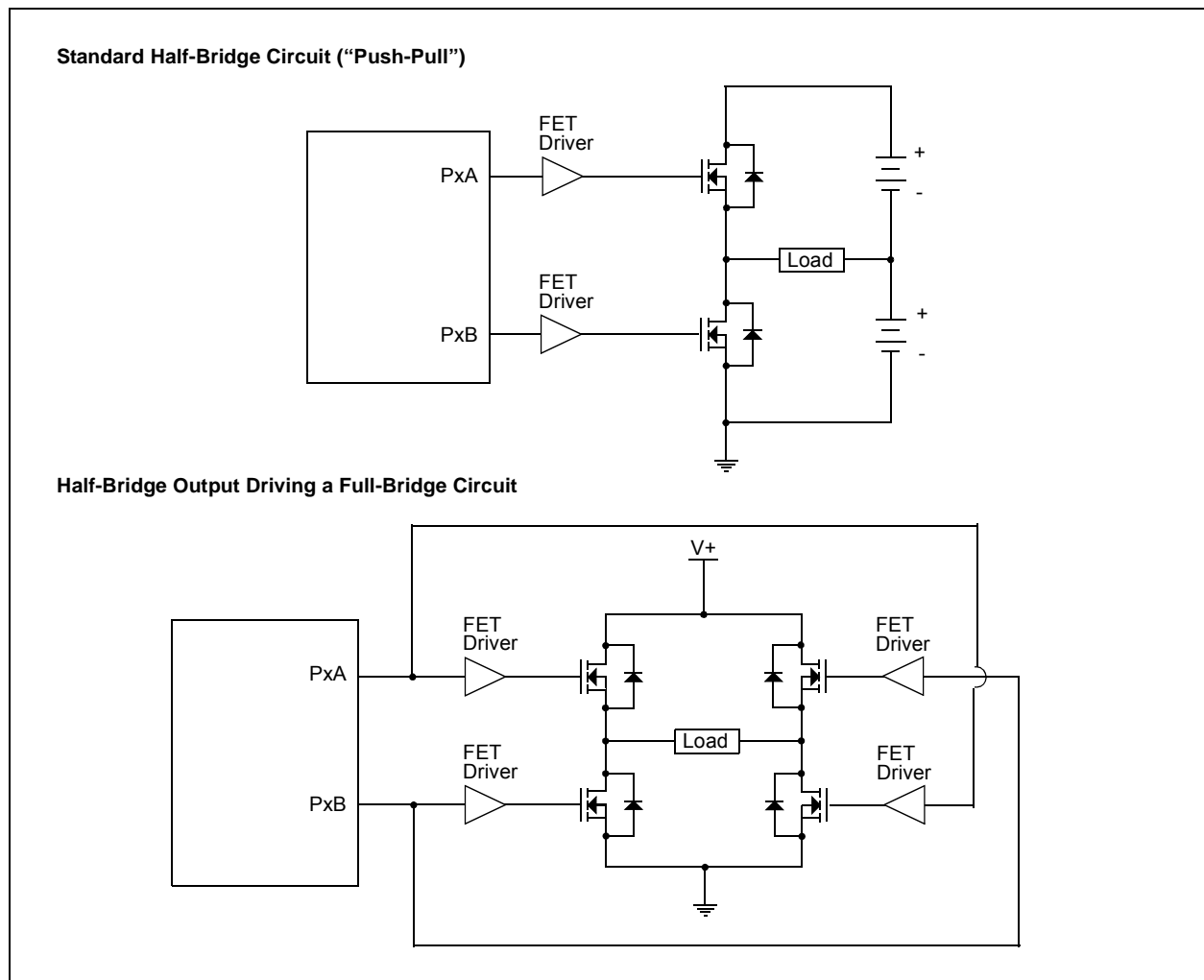


FIGURE 24-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS



25.5.5 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCLx line low effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCLx.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. Setting CKP will release SCLx and allow more communication.

25.5.5.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$ if the R/W bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the ACK sequence. Once the slave is ready, CKP is set by software and communication resumes.

Note 1: The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCLx.

2: Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCLx. It is now always cleared for read requests.

25.5.5.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set, the clock is always stretched. This is the only time the SCLx is stretched without CKP being cleared. SCLx is released immediately after a write to SSPxADD.

Note: Previous versions of the module did not stretch the clock if the second address byte did not match.

25.5.5.3 Byte NACKing

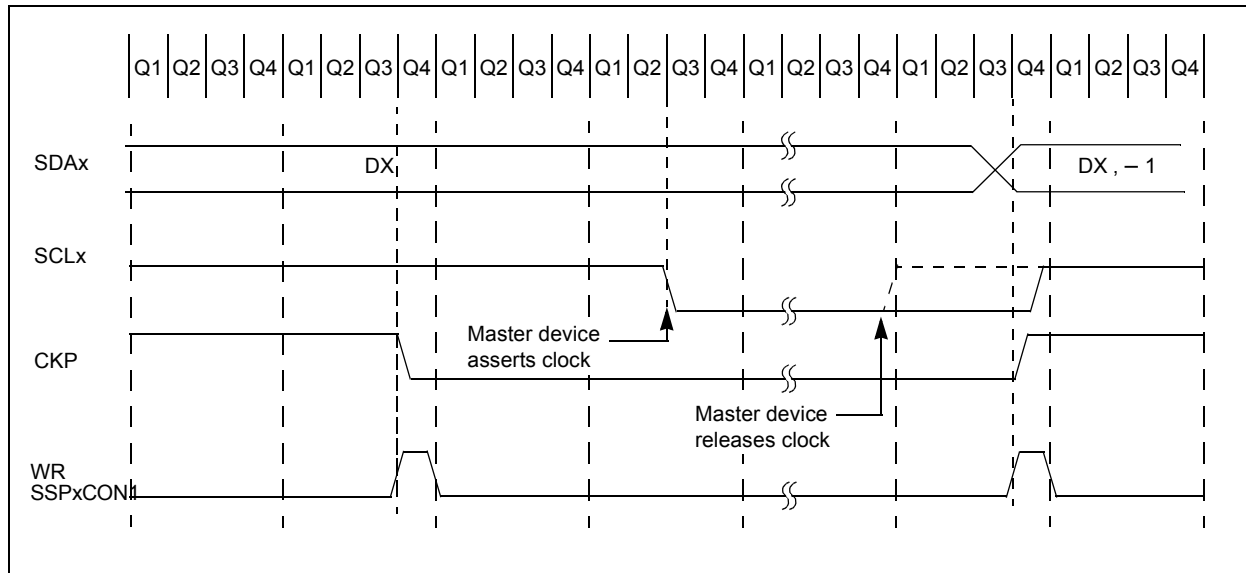
When AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCLx for a received matching address byte. When DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCLx for received data.

Stretching after the eighth falling edge of SCLx allows the slave to look at the received address or data and decide if it wants to ACK the received data.

25.5.6 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. However, clearing the CKP bit will not assert the SCLx output low until the SCLx output is already sampled low. Therefore, the CKP bit will not assert the SCLx line until an external I²C master device has already asserted the SCLx line. The SCLx output will remain low until the CKP bit is set and all other devices on the I²C bus have released SCLx. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCLx (see Figure 25-23).

FIGURE 25-23: CLOCK SYNCHRONIZATION TIMING



PIC16(L)F1825/9

26.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as V_{DD} or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 5.2.2 “Internal Clock Sources”** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 26.3.1 “Auto-Baud Detect”**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

REGISTER 26-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

- bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit⁽¹⁾
1 = Transmit enabled
0 = Transmit disabled
- bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode
- bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
Synchronous mode:
Don't care
- bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode
- bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full
- bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

REGISTER 26-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER ⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-x/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'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

- 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.

PIC16(L)F1825/9

29.0 INSTRUCTION SET SUMMARY

Each PIC16 instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 29-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of four oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

29.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 29-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 29-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
\overline{TO}	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
\overline{PD}	Power-down bit

30.1 DC Characteristics: PIC16(L)F1825/9-I/E (Industrial, Extended)

PIC16LF1825/9		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
PIC16F1825/9		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
D001	VDD	Supply Voltage (VDDMIN, VDDMAX)					
		PIC16LF1825/9	1.8 2.5	— —	3.6 3.6	V V	Fosc ≤ 16 MHz: Fosc ≤ 32 MHz (Note 2)
D001		PIC16F1825/9	1.8 2.5	— —	5.5 5.5	V V	Fosc ≤ 16 MHz: Fosc ≤ 32 MHz (Note 2)
D002*	VDR	RAM Data Retention Voltage ⁽¹⁾					
		PIC16LF1825/9	1.5	—	—	V	Device in Sleep mode
D002*		PIC16F1825/9	1.7	—	—	V	Device in Sleep mode
	VPOR*	Power-on Reset Release Voltage	—	1.6	—	V	
D002B*	VPORR*	Power-on Reset Rearm Voltage					
		PIC16LF1825/9	—	0.8	—	V	Device in Sleep mode
		PIC16F1825/9	—	1.5	—	V	Device in Sleep mode
D003	VADFVR	Fixed Voltage Reference Voltage for ADC	-8	—	6	%	1.024V, VDD ≥ 2.5V 2.048V, VDD ≥ 2.5V 4.096V, VDD ≥ 4.75V
D003A	VCDAFVR	Fixed Voltage Reference Voltage for Comparator and DAC	-11	—	7	%	1.024V, VDD ≥ 2.5V 2.048V, VDD ≥ 2.5V 4.096V, VDD ≥ 4.75V
D003C*	TCVFVR	Temperature Coefficient, Fixed Voltage Reference	—	-130	—	ppm/°C	
D003D*	$\Delta V_{FVR}/\Delta V_{IN}$	Line Regulation, Fixed Voltage Reference	—	0.270	—	%/V	
D004*	SVDD	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See Section 7.1 "Power-on Reset (POR)" for details.

* 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:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.
Note 2: PLL required for 32 MHz operation.

FIGURE 31-1: I_{DD} , LP OSCILLATOR MODE ($F_{osc} = 32$ kHz), PIC16LF1825/9 ONLY

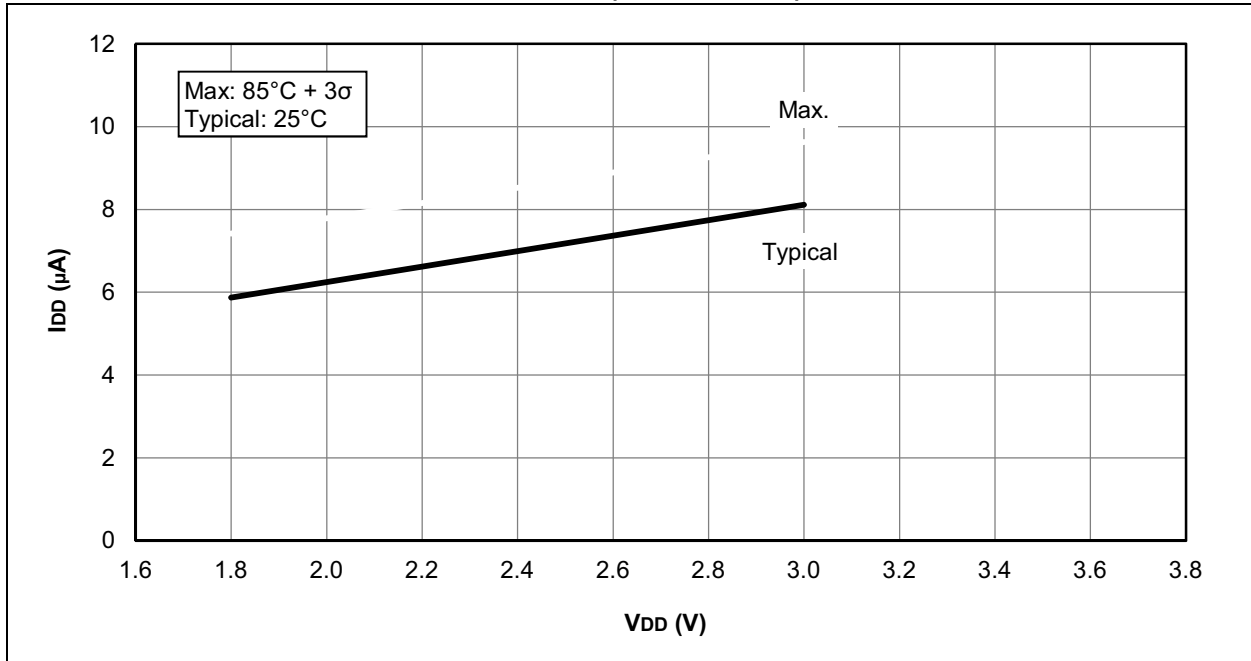
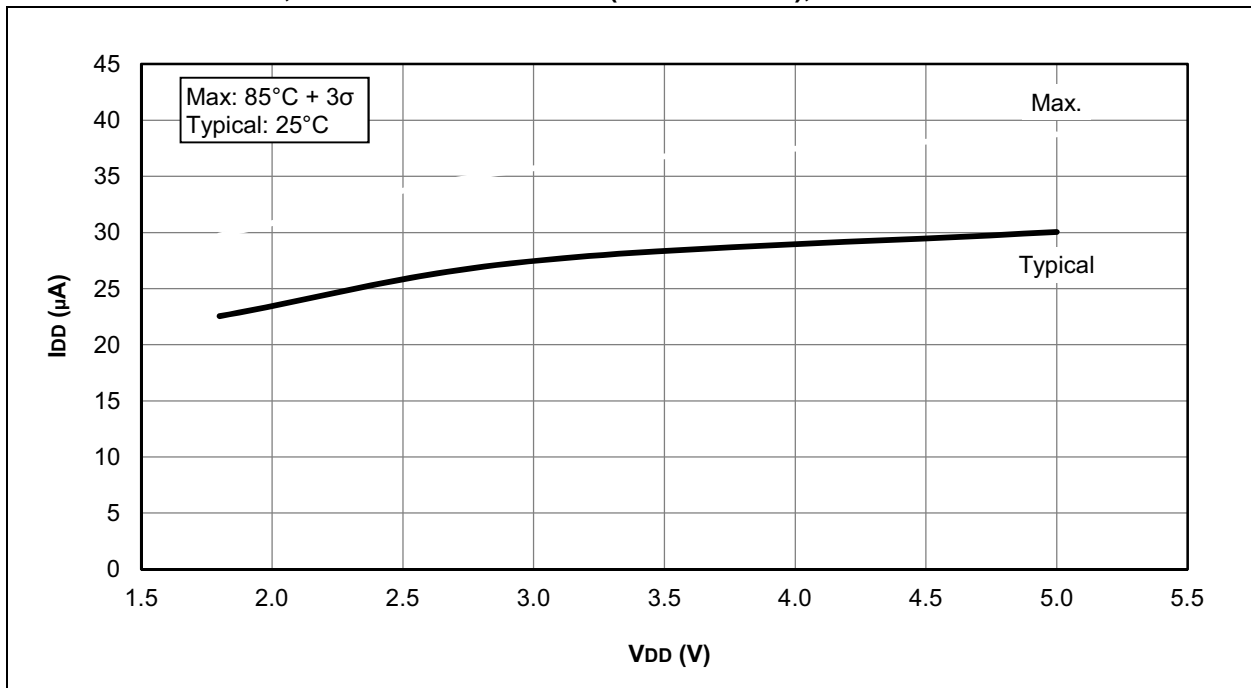


FIGURE 31-2: I_{DD} , LP OSCILLATOR MODE ($F_{osc} = 32$ kHz), PIC16F1825/9 ONLY



PIC16(L)F1825/9

FIGURE 31-23: I_{PD}, WATCHDOG TIMER (WDT), PIC16LF1825/9 ONLY

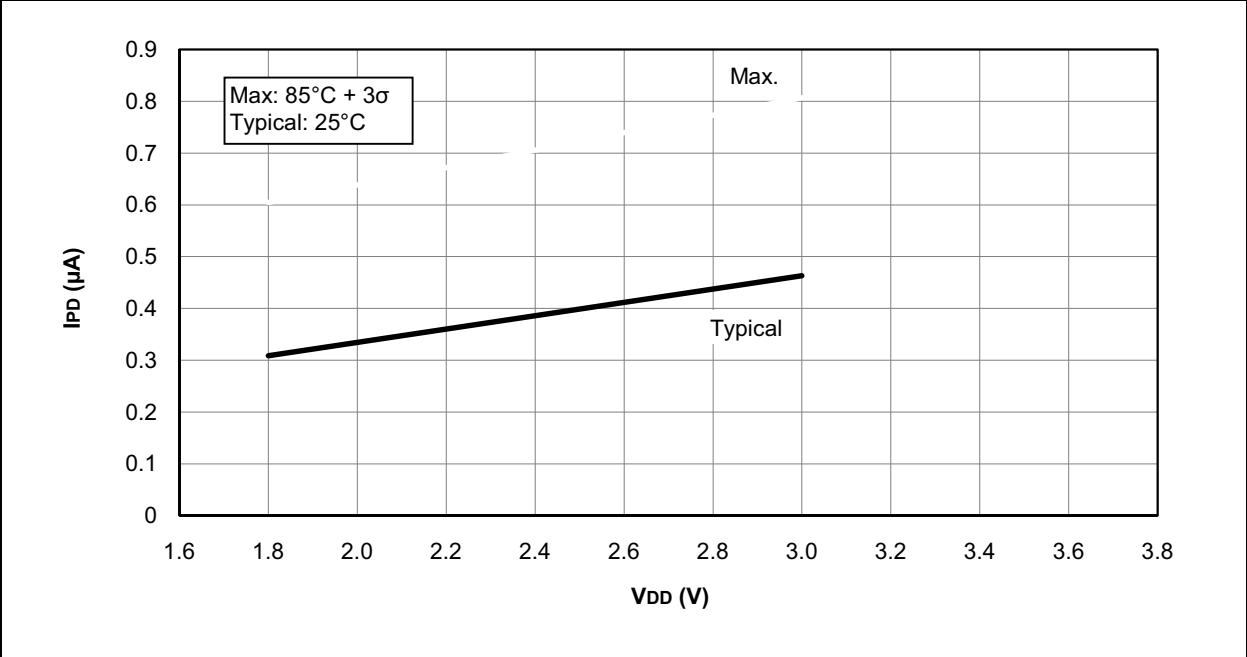


FIGURE 31-24: I_{PD}, WATCHDOG TIMER (WDT), PIC16F1825/9 ONLY

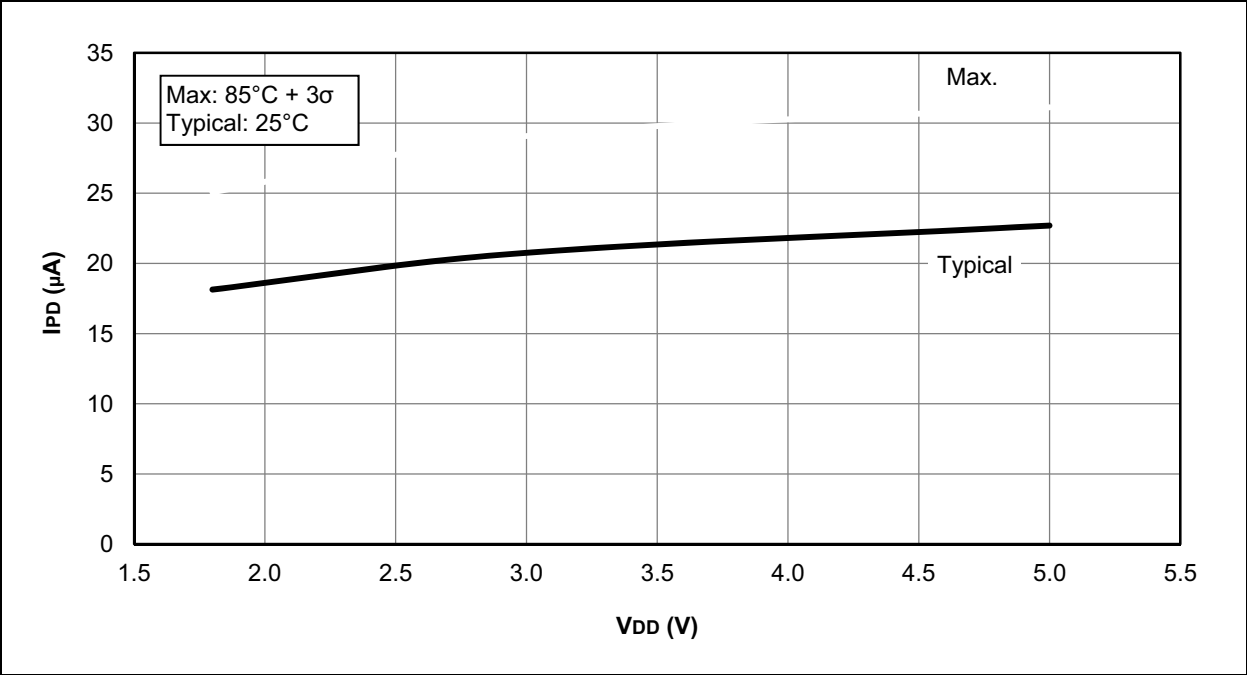


FIGURE 31-37: I_{PD} , COMPARATOR, LOW-POWER MODE ($CxSP = 0$), PIC16LF1825/9 ONLY

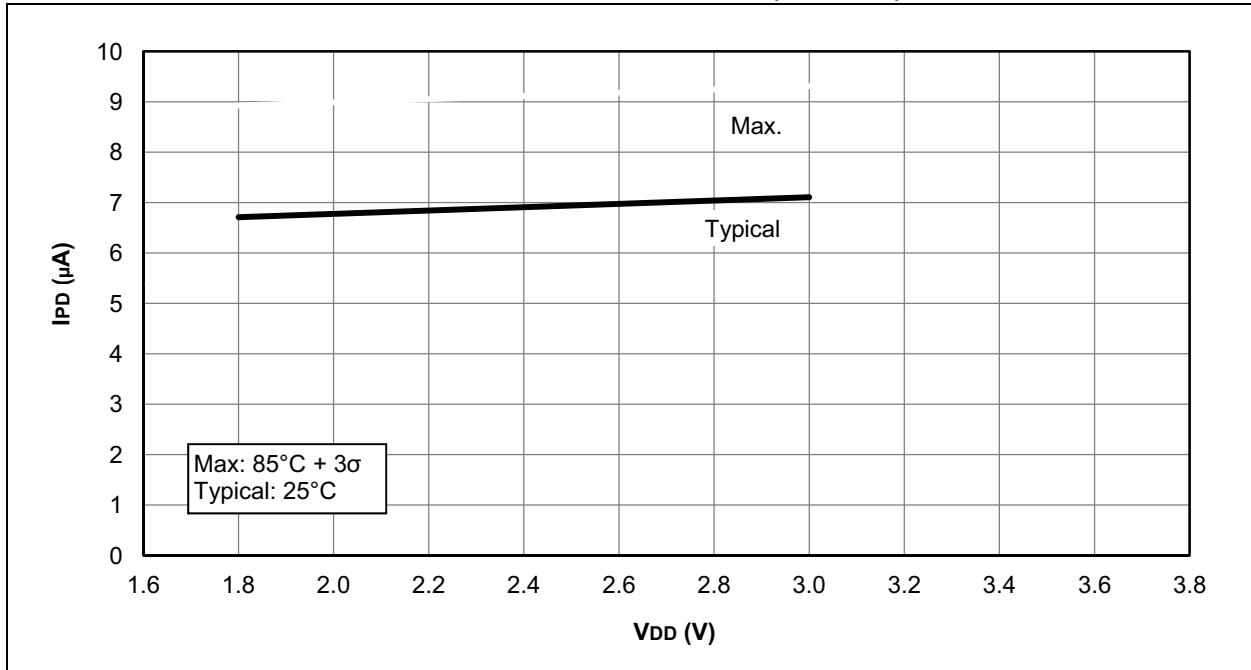
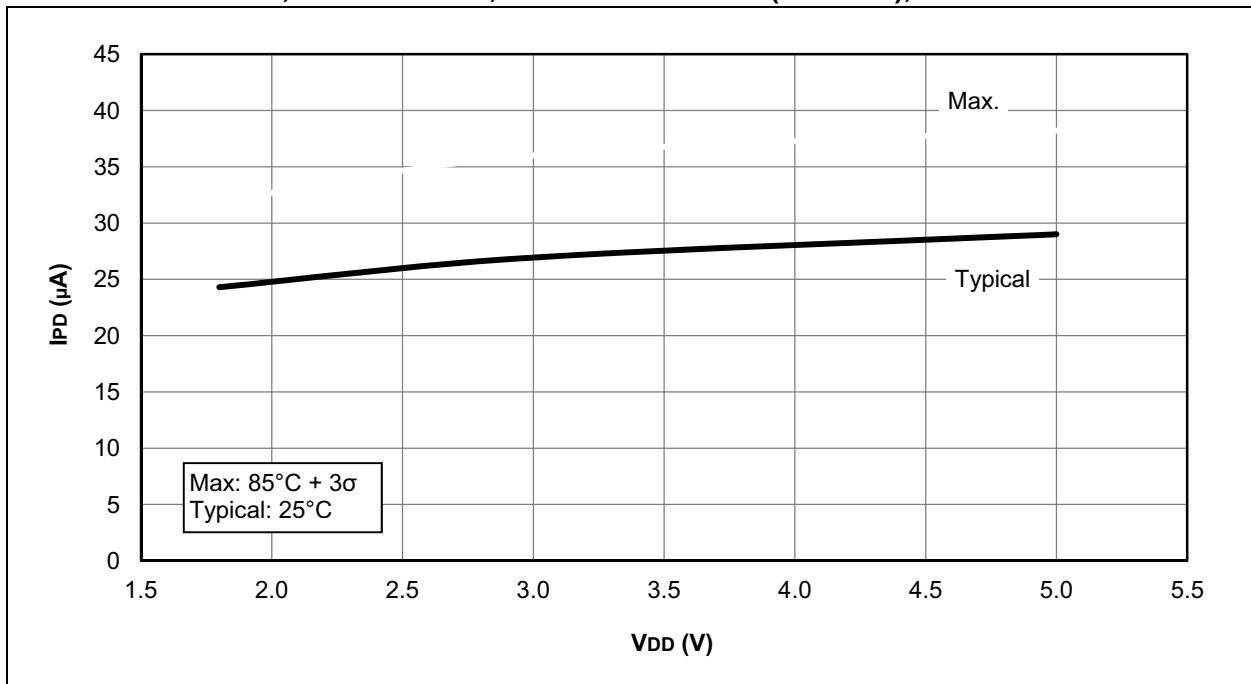


FIGURE 31-38: I_{PD} , COMPARATOR, LOW-POWER MODE ($CxSP = 0$), PIC16F1825/9 ONLY



PIC16(L)F1825/9

FIGURE 31-39: I_{PD}, COMPARATOR, NORMAL-POWER MODE (C_xSP = 1), PIC16LF1825/9 ONLY

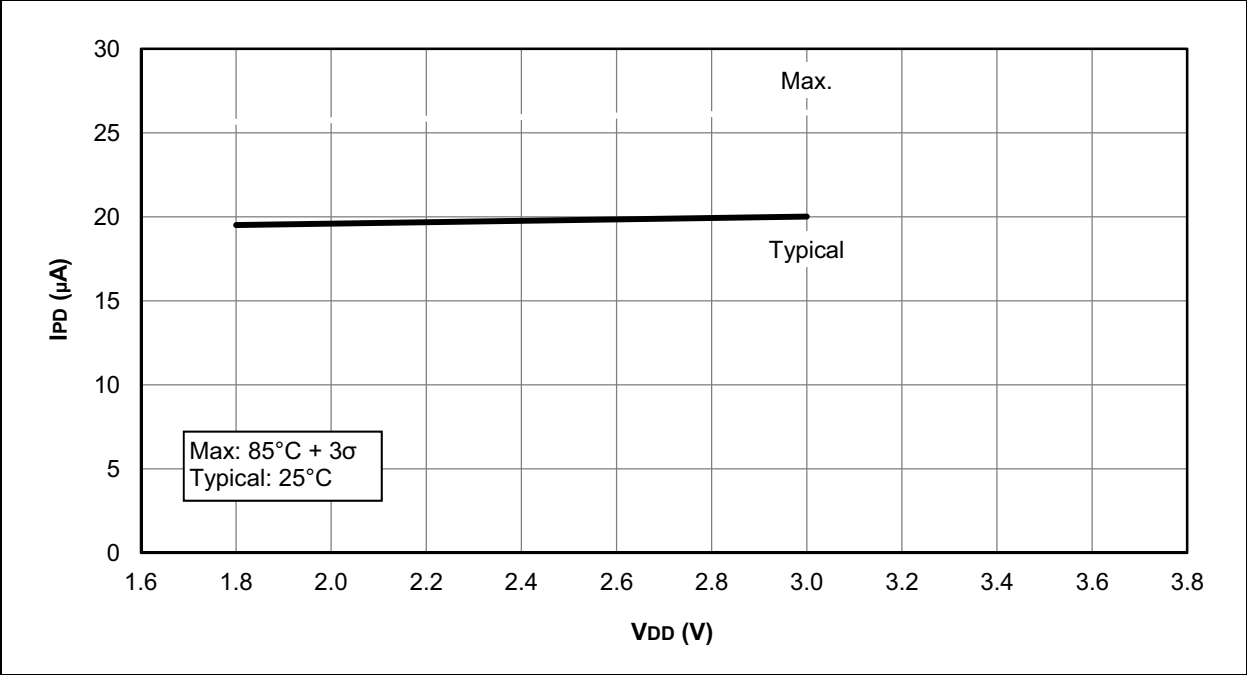


FIGURE 31-40: I_{PD}, COMPARATOR, NORMAL-POWER MODE (C_xSP = 1), PIC16F1825/9 ONLY

