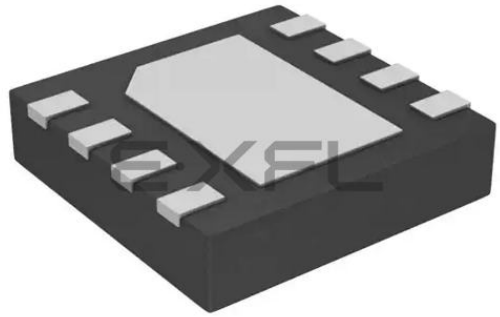


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

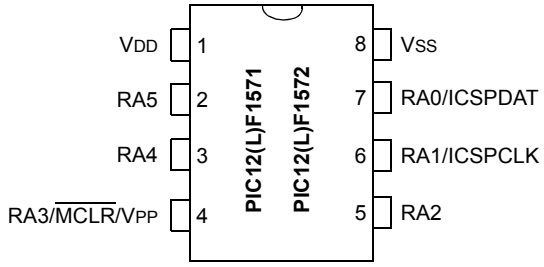
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	32MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	6
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 4x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-UDFN Exposed Pad
Supplier Device Package	8-UDFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12f1571-e-rf

PIN DIAGRAMS

Pin Diagram – 8-Pin PDIP, SOIC, DFN, MSOP, UDFN



Note: See Table 1 for location of all peripheral functions.

TABLE 1: 8-PIN ALLOCATION TABLE (PIC12(L)F1571/2)

I/O	8-Pin PDIP/SOIC/MSOP/DFN/UDFN	ADC	Reference	Comparator	Timers	PWM	EUSART(2)	CWG	Interrupt	Pull-up	Basic
RA0	7	AN0	DAC1OUT	C1IN+	—	PWM2	TX(2) CK(2)	CWG1B	IOC	Y	ICSPDAT ICDDAT
RA1	6	AN1	VREF+	C1IN0-	—	PWM1	RX(2) DT(2)	—	IOC	Y	ICSPCLK ICDCLK
RA2	5	AN2	—	C1OUT	T0CKI	PWM3	—	$\overline{\text{CWG1FLT}}$ CWG1A	IOC INT	Y	—
RA3	4	—	—	—	T1G(1)	—	—	—	IOC	Y	$\overline{\text{MCLR}}$ VPP
RA4	3	AN3	—	C1IN1-	T1G	PWM2(1)	TX(1,2) CK(1,2)	CWG1B(1)	IOC	Y	CLKOUT
RA5	2	—	—	—	T1CKI	PWM1(1)	RX(1,2) DT(1,2)	CWG1A(1)	IOC	Y	CLKIN
VDD	1	—	—	—	—	—	—	—	—	—	VDD
Vss	8	—	—	—	—	—	—	—	—	—	Vss

Note 1: Alternate pin function selected with the APFCON (Register 11-1) register.
2: PIC12(L)F1572 only.

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

3.2.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

3.2.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

EXAMPLE 3-1: RETLW INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available, so the older table read method must be used.

3.2.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRnH register and reading the matching INDFn register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDFn registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH operator will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
  DW     DATA0         ;First constant
  DW     DATA1         ;Second constant
  DW     DATA2
  DW     DATA3
my_function
  ;… LOTS OF CODE…
  MOVLW  DATA_INDEX
  ADDLW  LOW constants
  MOVWF  FSR1L
  MOVLW  HIGH constants ;MSb is set
                         automatically
  MOVWF  FSR1H
  BTFSC  STATUS,C      ;carry from ADDLW?
  INCF   FSR1H,f       ;yes
  MOVIW  0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

TABLE 3-5: PIC12(L)F1571/2 MEMORY MAP, BANK 8-23

BANK 8	BANK 9	BANK 10	BANK 11	BANK 12	BANK 13	BANK 14	BANK 15
400h Core Registers (Table 3-2) 40Bh	480h Core Registers (Table 3-2) 48Bh	500h Core Registers (Table 3-2) 50Bh	580h Core Registers (Table 3-2) 58Bh	600h Core Registers (Table 3-2) 60Bh	680h Core Registers (Table 3-2) 68Bh	700h Core Registers (Table 3-2) 70Bh	780h Core Registers (Table 3-2) 78Bh
40Ch —	48Ch —	50Ch —	58Ch —	60Ch —	68Ch —	70Ch —	78Ch —
40Dh —	48Dh —	50Dh —	58Dh —	60Dh —	68Dh —	70Dh —	78Dh —
40Eh —	48Eh —	50Eh —	58Eh —	60Eh —	68Eh —	70Eh —	78Eh —
40Fh —	48Fh —	50Fh —	58Fh —	60Fh —	68Fh —	70Fh —	78Fh —
410h —	490h —	510h —	590h —	610h —	690h —	710h —	790h —
411h —	491h —	511h —	591h —	611h —	691h CWG1DBR	711h —	791h —
412h —	492h —	512h —	592h —	612h —	692h CWG1DBF	712h —	792h —
413h —	493h —	513h —	593h —	613h —	693h CWG1CON0	713h —	793h —
414h —	494h —	514h —	594h —	614h —	694h CWG1CON1	714h —	794h —
415h —	495h —	515h —	595h —	615h —	695h CWG1CON2	715h —	795h —
416h —	496h —	516h —	596h —	616h —	696h —	716h —	796h —
417h —	497h —	517h —	597h —	617h —	697h —	717h —	797h —
418h —	498h —	518h —	598h —	618h —	698h —	718h —	798h —
419h —	499h —	519h —	599h —	619h —	699h —	719h —	799h —
41Ah —	49Ah —	51Ah —	59Ah —	61Ah —	69Ah —	71Ah —	79Ah —
41Bh —	49Bh —	51Bh —	59Bh —	61Bh —	69Bh —	71Bh —	79Bh —
41Ch —	49Ch —	51Ch —	59Ch —	61Ch —	69Ch —	71Ch —	79Ch —
41Dh —	49Dh —	51Dh —	59Dh —	61Dh —	69Dh —	71Dh —	79Dh —
41Eh —	49Eh —	51Eh —	59Eh —	61Eh —	69Eh —	71Eh —	79Eh —
41Fh —	49Fh —	51Fh —	59Fh —	61Fh —	69Fh —	71Fh —	79Fh —
420h Unimplemented Read as '0' 46Fh	4A0h Unimplemented Read as '0' 4EFh	520h Unimplemented Read as '0' 56Fh	5A0h Unimplemented Read as '0' 5EFh	620h Unimplemented Read as '0' 66Fh	6A0h Unimplemented Read as '0' 6EFh	720h Unimplemented Read as '0' 76Fh	7A0h Unimplemented Read as '0' 7EFh
470h Accesses 70h-7Fh 47Fh	4F0h Accesses 70h-7Fh 4FFh	570h Accesses 70h-7Fh 57Fh	5F0h Accesses 70h-7Fh 5FFh	670h Accesses 70h-7Fh 67Fh	6F0h Accesses 70h-7Fh 6FFh	770h Accesses 70h-7Fh 77Fh	7F0h Accesses 70h-7Fh 7FFh

BANK 16	BANK 17	BANK 18	BANK 19	BANK 20	BANK 21	BANK 22	BANK 23
800h Core Registers (Table 3-2) 80Bh	880h Core Registers (Table 3-2) 88Bh	900h Core Registers (Table 3-2) 90Bh	980h Core Registers (Table 3-2) 98Bh	A00h Core Registers (Table 3-2) A0Bh	A80h Core Registers (Table 3-2) A8Bh	B00h Core Registers (Table 3-2) B0Bh	B80h Core Registers (Table 3-2) B8Bh
80Ch Unimplemented Read as '0' 86Fh	88Ch Unimplemented Read as '0' 8EFh	90Ch Unimplemented Read as '0' 96Fh	98Ch Unimplemented Read as '0' 9EFh	A0Ch Unimplemented Read as '0' A6Fh	A8Ch Unimplemented Read as '0' AEFh	B0Ch Unimplemented Read as '0' B6Fh	B8Ch Unimplemented Read as '0' BEFh
870h Accesses 70h-7Fh 87Fh	8F0h Accesses 70h-7Fh 8FFh	970h Accesses 70h-7Fh 97Fh	9F0h Accesses 70h-7Fh 9FFh	A70h Accesses 70h-7Fh A7Fh	AF0h Accesses 70h-7Fh AFFh	B70h Accesses 70h-7Fh B7Fh	BF0h Accesses 70h-7Fh BFFh

Legend: ☐ = Unimplemented data memory locations, read as '0'.

5.2.2.3 Internal Oscillator Frequency Adjustment

The 500 kHz internal oscillator is factory calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 5-3). Since the HFINTOSC and MFINTOSC clock sources are derived from the 500 kHz internal oscillator, a change in the OSCTUNE register value will apply to both.

The default value of the OSCTUNE register is '0'. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depends on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT) and peripherals, are *not* affected by the change in frequency.

5.2.2.4 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a multiplexer (see Figure 5-1). Select 31 kHz, via software, using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<3:0> (OSCCON<6:3>) = 0000) as the system clock source (SCS<1:0> (OSCCON<1:0>) = 1x) or when any of the following are enabled:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired LF frequency, and
- Set FOSC<1:0> = 00, or
- Set the System Clock Source x (SCSx) bits of the OSCCON register to '1x'

Peripherals that use the LFINTOSC are:

- Power-up Timer (PWRT)
- Watchdog Timer (WDT)

The Low-Frequency Internal Oscillator Ready bit (LFIOFR) of the OSCSTAT register indicates when the LFINTOSC is running.

5.2.2.5 FRC

The FRC clock is an uncalibrated, nominal 600 kHz peripheral clock source.

The FRC is automatically turned on by the peripherals requesting the FRC clock.

The FRC clock will continue to run during Sleep.

5.2.2.6 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The postscaler outputs of the 16 MHz HFINTOSC, **500 kHz MFINTOSC** and **31 kHz** LFINTOSC output connect to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits, IRCF<3:0> of the OSCCON register, select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 32 MHz (requires 4x PLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

Note:	Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCFx bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

REGISTER 7-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	U-0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
—	—	C1IE	—	—	—	—	—
bit 7							bit 0

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

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

bit 5 **C1IE:** Comparator C1 Interrupt Enable bit
1 = Enables the Comparator C1 interrupt
0 = Disables the Comparator C1 interrupt

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

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

See Table 10-1 for erase row size and the number of write latches for Flash program memory.

TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC12(L)F1571	16	16
PIC12(L)F1572		

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

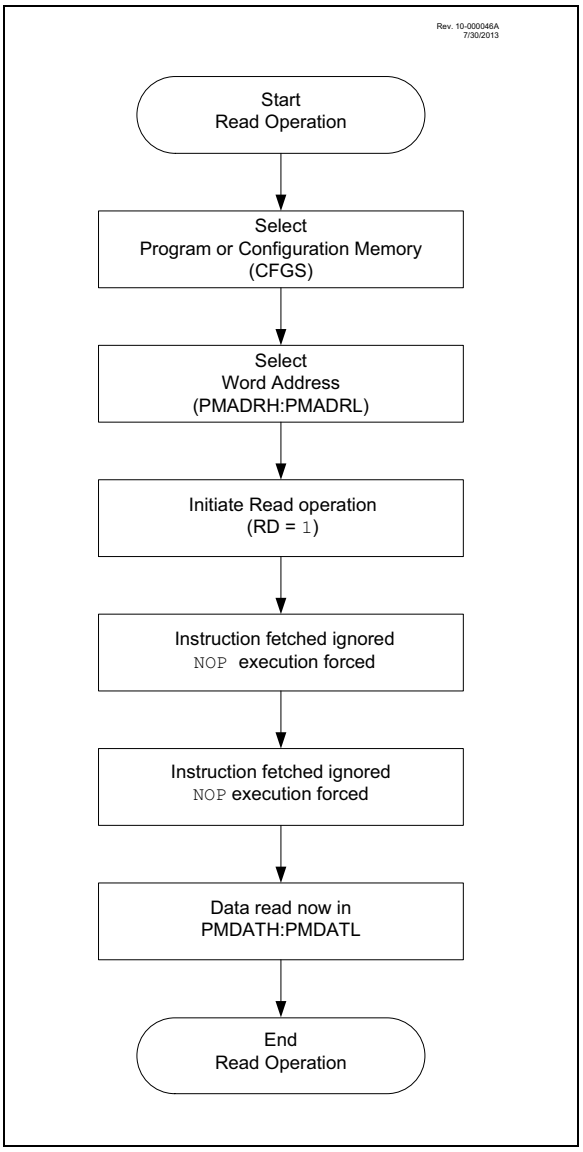
1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit, RD, of the PMCON1 register.

Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF PMCON1,RD" instruction to be ignored. The data is available in the very next cycle in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

The PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

Note:	The two instructions following a program memory read are required to be NOPs. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART



11.0 I/O PORTS

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)
- INLVLx (Input Level Control)
- ODCONx registers (Open-Drain Control)
- SLRCONx registers (Slew Rate Control)

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

- ANSELx (Analog Select)
- WPUx (Weak Pull-up)

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.

TABLE 11-1: PORT AVAILABILITY PER DEVICE

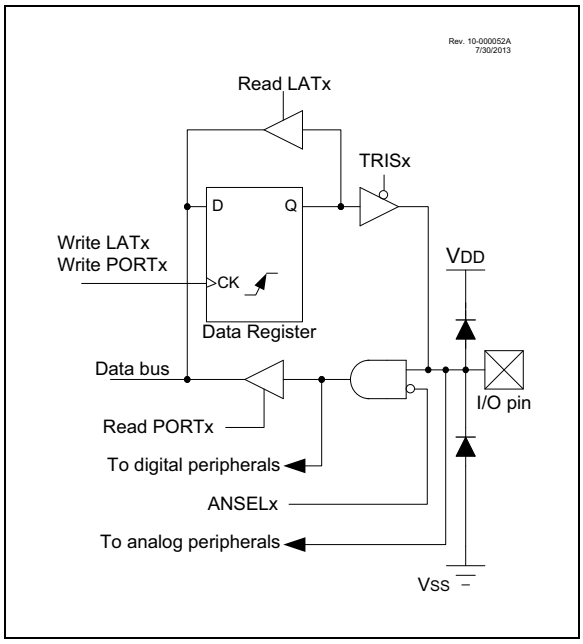
Device	PORTA
PIC12(L)F1571	•
PIC12(L)F1572	•

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 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 ANSELx 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 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



11.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 11-1. For this device family, the following functions can be moved between different pins.

- RX/DT
- TX/CK
- CWGOUTA
- CWGOUTB
- PWM2
- PWM1

These bits have no effect on the values of any TRISx register. PORTx and TRISx overrides will be routed to the correct pin. The unselected pin will be unaffected.

11.2 Register Definitions: Alternate Pin Function Control

REGISTER 11-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RXDTSEL	CWGASEL	CWGBSEL	—	T1GSEL	TXCKSEL	P2SEL	P1SEL
bit 7							bit 0

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

bit 7 **RXDTSEL:** Pin Selection bit
- 1 = RX/DT function is on RA5
- 0 = RX/DT function is on RA1

bit 6 **CWGASEL:** Pin Selection bit
- 1 = CWGOUTA function is on RA5
- 0 = CWGOUTA function is on RA2

bit 5 **CWGBSEL:** Pin Selection bit
- 1 = CWGOUTB function is on RA4
- 0 = CWGOUTB function is on RA0

bit 4 **Unimplemented:** Read as ‘0’

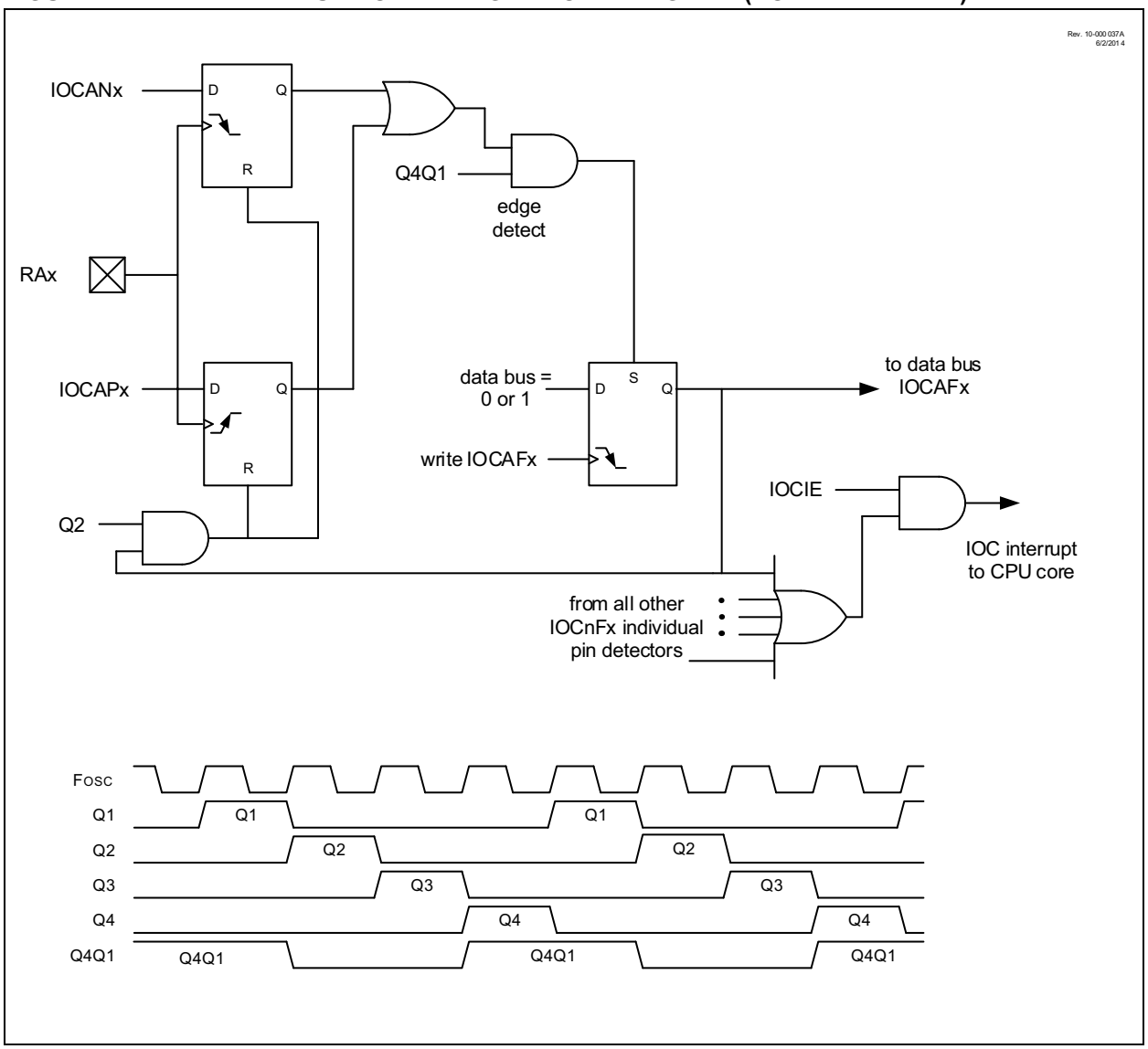
bit 3 **T1GSEL:** Pin Selection bit
- 1 = T1G function is on RA3
- 0 = T1G function is on RA4

bit 2 **TXCKSEL:** Pin Selection bit
- 1 = TX/CK function is on RA4
- 0 = TX/CK function is on RA0

bit 1 **P2SEL:** Pin Selection bit
- 1 = PWM2 function is on RA4
- 0 = PWM2 function is on RA0

bit 0 **P1SEL:** Pin Selection bit
- 1 = PWM1 function is on RA5
- 0 = PWM1 function is on RA1

FIGURE 12-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)



16.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DACR<4:0> bits of the DACxCON1 register.

The DAC output voltage can be determined by using Equation 16-1.

16.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 26-16.

16.3 DAC Voltage Reference Output

The unbuffered DAC voltage can be output to the DACxOUTn pin(s) by setting the respective DACOEn bit(s) of the DACxCON0 register. Selecting the DAC reference voltage for output on either DACxOUTn pin automatically overrides the digital output buffer, the weak pull-up and digital input threshold detector functions of that pin.

Reading the DACxOUTn pin when it has been configured for DAC reference voltage output will always return a '0'.

Note:	The unbuffered DAC output (DACxOUTn) is not intended to drive an external load.

16.4 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DACxCON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

16.5 Effects of a Reset

A device Reset affects the following:

- DACx is disabled.
- DACx output voltage is removed from the DACxOUTn pin(s).
- The DACR<4:0> range select bits are cleared.

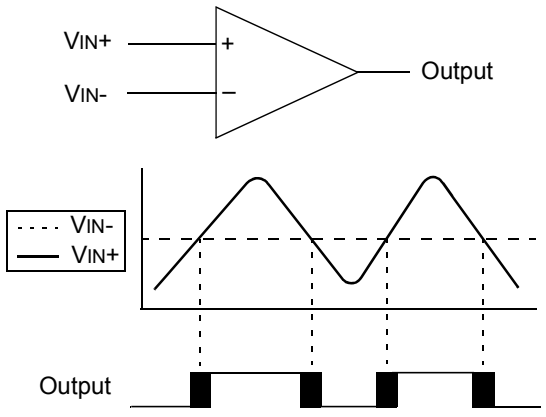
EQUATION 16-1: DAC OUTPUT VOLTAGE

IF DACEN = 1

$$DACx\_output = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[4:0]}{2^5} \right) + V_{SOURCE-}$$

Note: See the DACxCON0 register for the available $V_{SOURCE+}$ and $V_{SOURCE-}$ selections.

FIGURE 17-2: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

17.2 Comparator Control

The comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 17-1) contains control and status bits for the following:

- Enable
- Output selection
- Output polarity
- Speed/power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 register (see Register 17-2) contains control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

17.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

17.2.2 COMPARATOR POSITIVE INPUT SELECTION

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN+ analog pin
- DAC1_output
- FVR_buffer2
- VSS

See **Section 13.0 "Fixed Voltage Reference (FVR)"** for more information on the Fixed Voltage Reference module.

See **Section 16.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

17.2.3 COMPARATOR NEGATIVE INPUT SELECTION

The CxNCH<2:0> bits of the CMxCON0 register direct one of the input sources to the comparator inverting input.

Note: To use CxIN+ and CxIN- pins as analog input, the appropriate bits must be set in the ANSELx register and the corresponding TRISx bits must also be set to disable the output drivers.

17.2.4 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- CxOE bit of the CMxCON0 register must be set
- Corresponding TRISx bit must be cleared
- CxON bit of the CMxCON0 register must be set

The synchronous comparator output signal (CxOUT_sync) is available to the following peripheral(s):

- Analog-to-Digital Converter (ADC)
- Timer1

The asynchronous comparator output signal (CxOUT_async) is available to the following peripheral(s):

- Complementary Waveform Generator (CWG)

Note 1: The CxOE bit of the CMxCON0 register overrides the port data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override.

2: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

20.5 Register Definitions: Timer2 Control

REGISTER 20-1: T2CON: TIMER2 CONTROL REGISTER

U-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
—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
bit 7							bit 0

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

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

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscaler Select bits
0000 = 1:1 Postscaler
0001 = 1:2 Postscaler
0010 = 1:3 Postscaler
0011 = 1:4 Postscaler
0100 = 1:5 Postscaler
0101 = 1:6 Postscaler
0110 = 1:7 Postscaler
0111 = 1:8 Postscaler
1000 = 1:9 Postscaler
1001 = 1:10 Postscaler
1010 = 1:11 Postscaler
1011 = 1:12 Postscaler
1100 = 1:13 Postscaler
1101 = 1:14 Postscaler
1110 = 1:15 Postscaler
1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
10 = Prescaler is 16
11 = Prescaler is 64

TABLE 20-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

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	IOCIE	TMR0IF	INTF	IOCIF	74
PIE1	TMR1GIE	ADIE	RCIE(1)	TXIE(1)	—	—	TMR2IE	TMR1IE	75
PIR1	TMR1GIF	ADIF	RCIF(1)	TXIF(1)	—	—	TMR2IF	TMR1IF	78
PR2	Timer2 Module Period Register								171*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		173
TMR2	Holding Register for the 8-bit TMR2 Count								171*

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Timer2 module.
* Page provides register information.

Note 1: PIC12(L)F1572 only.

21.4 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH/SPBRGL register pair determines the period of the free-running baud rate timer. In Asynchronous mode, the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Table 21-3 contains the formulas for determining the baud rate. Example 21-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various Asynchronous modes have been computed for your convenience and are shown in Table 21-3. It may be advantageous to use the high baud rate (BRGH = 1) or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH/SPBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

EXAMPLE 21-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$Desired\ Baud\ Rate = \frac{FOSC}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SPBRGH:SPBRGL:

$$X = \frac{\frac{FOSC}{Desired\ Baud\ Rate}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$Calculated\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$Error = \frac{Calc.\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

REGISTER 22-7: PWMxPHH: PWMx PHASE COUNT HIGH 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
PH<15:8>							
bit 7							bit 0

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

bit 7-0 **PH<15:8>**: PWMx Phase High bits
Upper eight bits of PWM phase count.

REGISTER 22-8: PWMxPHL: PWMx PHASE COUNT LOW 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
PH<7:0>							
bit 7							bit 0

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

bit 7-0 **PH<7:0>**: PWMx Phase Low bits
Lower eight bits of PWM phase count.

23.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces a complementary waveform with dead-band delay from a selection of input sources.

The CWG module has the following features:

- Selectable dead-band clock source control
- Selectable input sources
- Output enable control
- Output polarity control
- Dead-band control with independent 6-bit rising and falling edge dead-band counters
- Auto-shutdown control with:
  - Selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control

23.1 Fundamental Operation

The CWG generates two output waveforms from the selected input source.

The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby, creating a time delay immediately where neither output is driven. This is referred to as dead time and is covered in **Section 23.5 "Dead-Band Control"**. A typical operating waveform with dead band, generated from a single input signal, is shown in Figure 23-2.

It may be necessary to guard against the possibility of circuit Faults or a feedback event arriving too late, or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is covered in **Section 23.9 "Auto-Shutdown Control"**.

23.2 Clock Source

The CWG module allows the following clock sources to be selected:

- FOSC (system clock)
- HFINTOSC (16 MHz only)

The clock sources are selected using the G1CS0 bit of the CWGxCON0 register (Register 23-1).

23.3 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 23-1.

TABLE 23-1: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name
Comparator C1	C1OUT_sync
PWM1	PWM1_output
PWM2	PWM2_output
PWM3	PWM3_output

The input sources are selected using the GxIS<2:0> bits in the CWGxCON1 register (Register 23-2).

23.4 Output Control

Immediately after the CWG module is enabled, the complementary drive is configured with both CWGxA and CWGxB drives cleared.

23.4.1 OUTPUT ENABLES

Each CWG output pin has individual output enable control. Output enables are selected with the GxOEA and GxOEB bits of the CWGxCON0 register. When an output enable control is cleared, the module asserts no control over the pin. When an output enable is set, the override value or active PWM waveform is applied to the pin per the port priority selection. The output pin enables are dependent on the module enable bit, GxEN. When GxEN is cleared, CWG output enables and CWG drive levels have no effect.

23.4.2 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the GxPOLA and GxPOLB bits of the CWGxCON0 register.

TABLE 23-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	114
CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	238
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		239
CWG1CON2	G1ASE	G1ARSEN	—	—	—	G1ASDSC1	G1ASDSFLT	—	240
CWG1DBF	—	—	CWG1DBF<5:0>						241
CWG1DBR	—	—	CWG1DBR<5:0>						241
TRISA	—	—	TRISA<5:4>		—(1)	TRISA2	TRISA<1:0>		113

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the CWG.

Note 1: Unimplemented, read as '1'.

FIGURE 27-9: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC12LF1571/2 ONLY

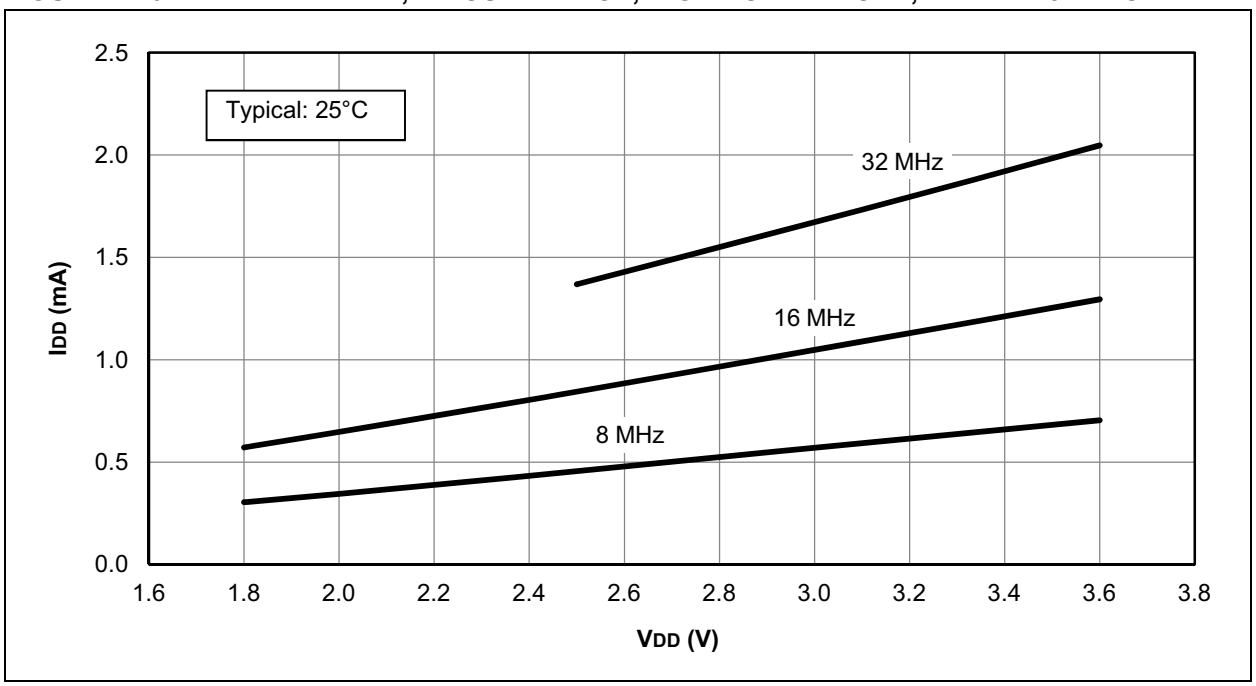


FIGURE 27-10: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC12LF1571/2 ONLY

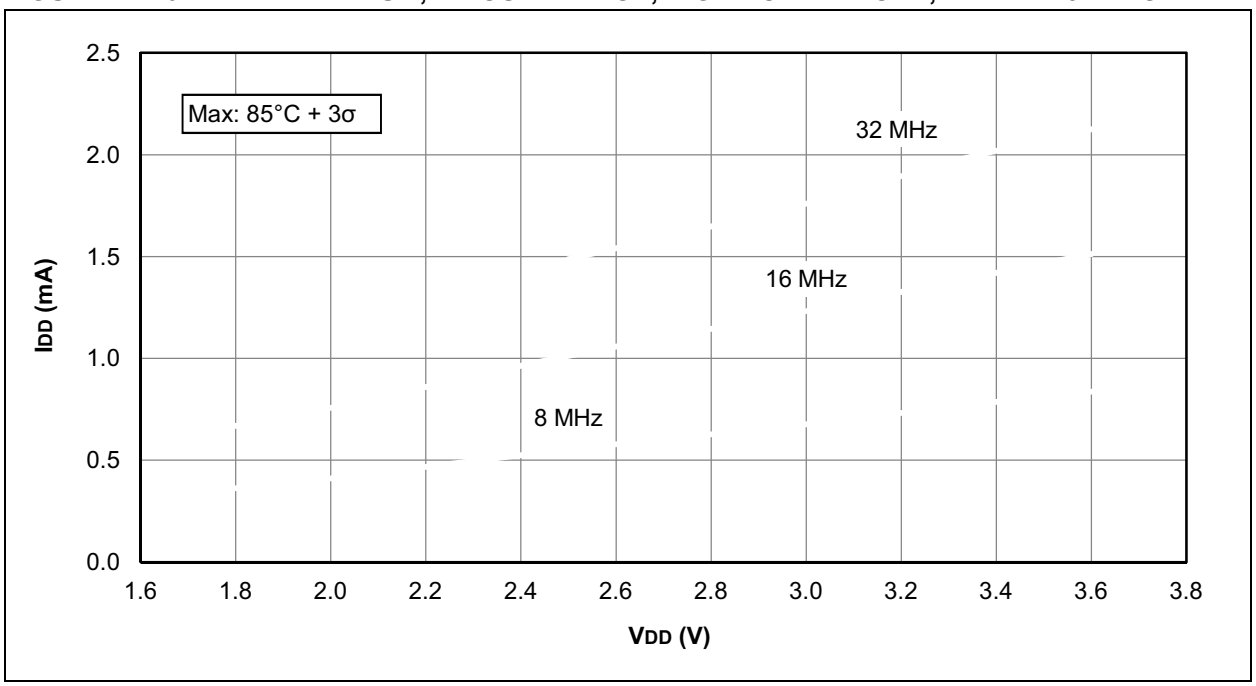


FIGURE 27-31: IPD, ADC NON-CONVERTING, PIC12LF1571/2 ONLY

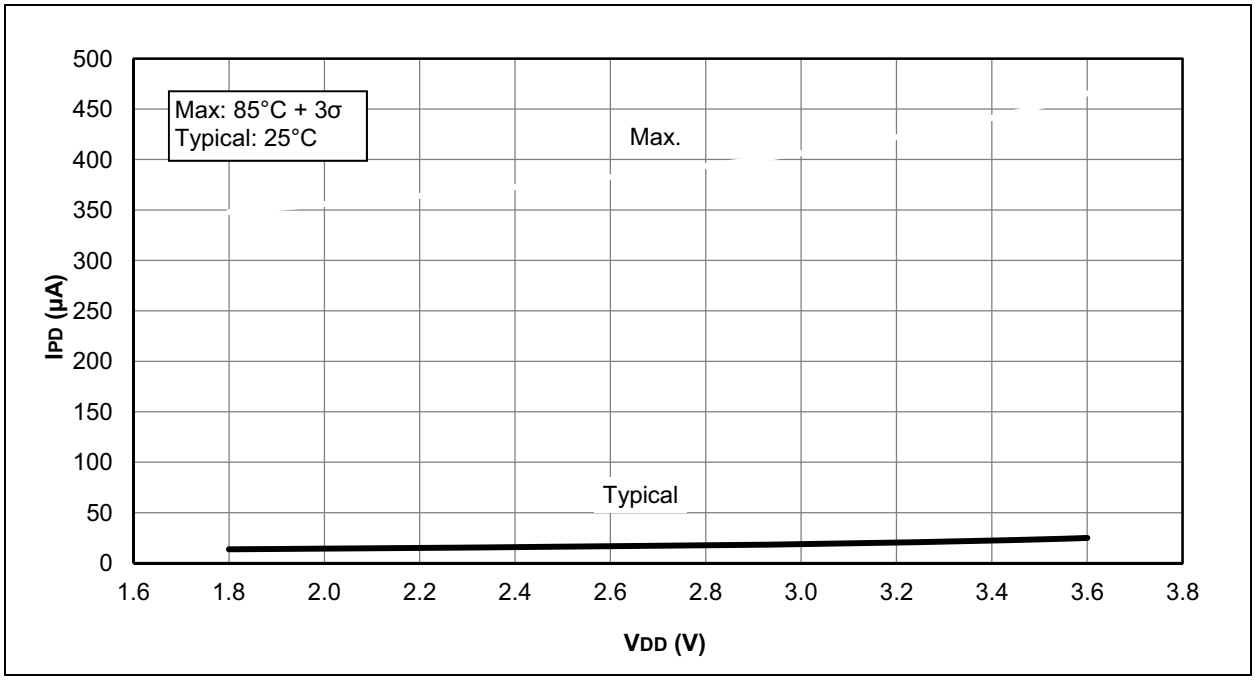
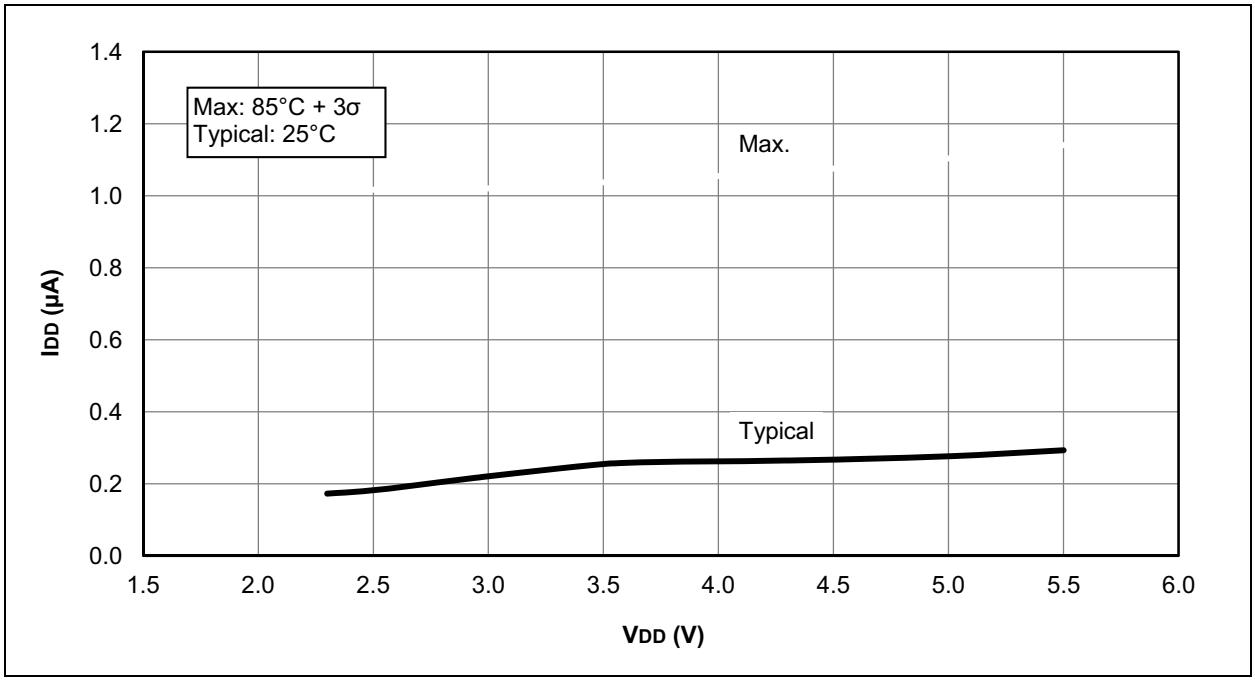


FIGURE 27-32: IPD, ADC NON-CONVERTING, PIC12F1571/2 ONLY



NOTES: