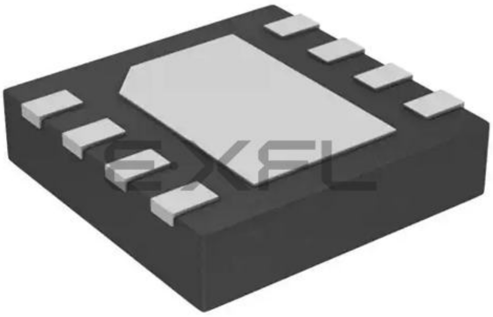
Microchip Technology - PIC12F1571-I/RF Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

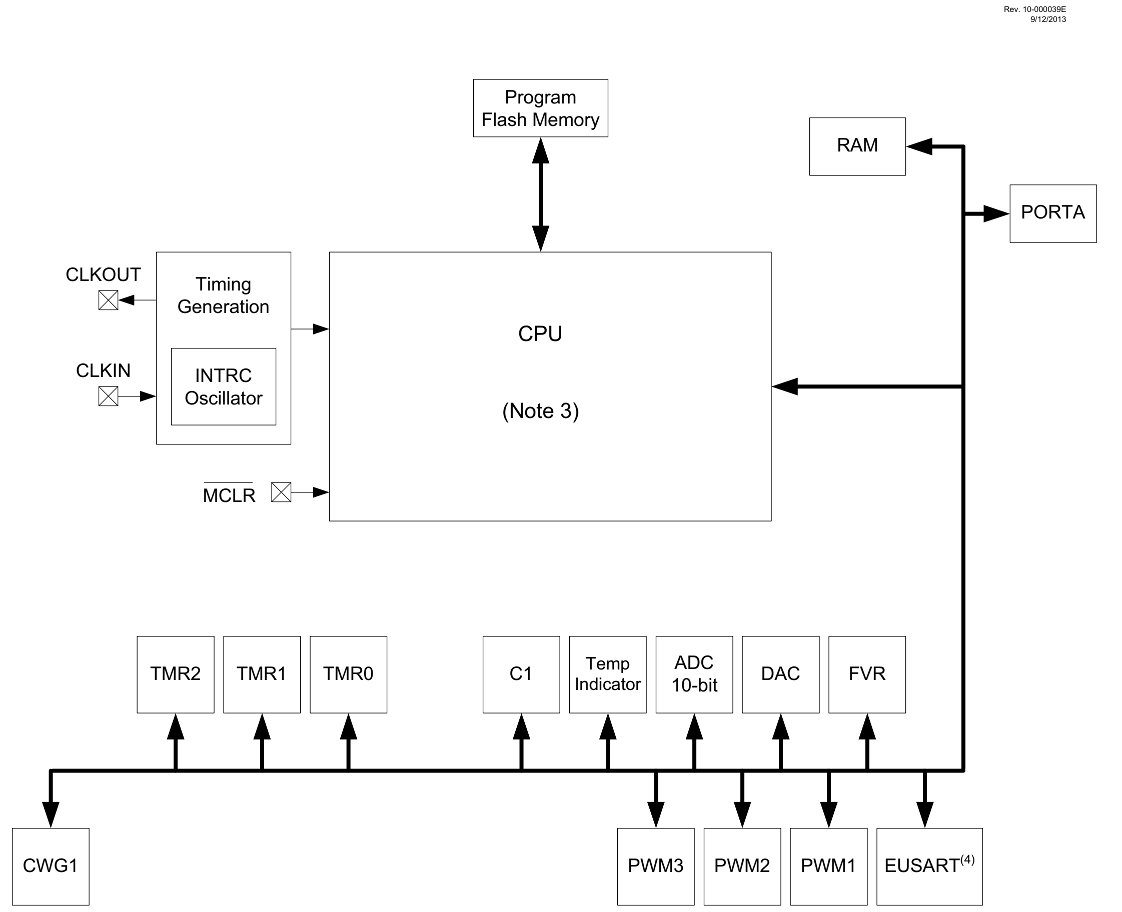
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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 ~ 85°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-i-rf

NOTES:

FIGURE 1-1: PIC12(L)F1571/2 BLOCK DIAGRAM



2: See Table 1-1 for peripherals available on specific devices.
3: See Figure 2-1.
4: PIC12(L)F1572 only.

TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other Resets
Bank 4											
20Ch	WPUA	—	—	WPUA<5:0>						`--11 1111`	`--11 1111`
20Dh	—	Unimplemented								—	—
20Eh to 21Fh	—	Unimplemented								—	—
Bank 5											
28Ch	ODCONA	—	—	ODA<5:4>		—	ODA<2:0>			`--11 -111`	`--11 -111`
28Dh to 29Fh	—	Unimplemented								—	—
Bank 6											
30Ch	SLRCONA	—	—	SLRA<5:4>		—	SLRA<2:0>			`--11 -111`	`--11 -111`
30Dh to 31Fh	—	Unimplemented								—	—
Bank 7											
38Ch	INLVLA	—	—	INLVLA<5:0>						`--11 1111`	`--11 1111`
38Dh to 390h	—	Unimplemented								—	—
391h	IOCAP	—	—	IOCAP<5:0>						`--00 0000`	`--00 0000`
392h	IOCAN	—	—	IOCAN<5:0>						`--00 0000`	`--00 0000`
393h	IOCAF	—	—	IOCAF<5:0>						`--00 0000`	`--00 0000`
394h to 39Fh	—	Unimplemented								—	—
Bank 8											
40Ch to 41Fh	—	Unimplemented								—	—
Bank 9											
48Ch to 49Fh	—	Unimplemented								—	—

Legend: x = unknown; u = unchanged; q = value depends on condition; — = unimplemented; `r` = reserved. Shaded locations are unimplemented, read as ‘`0`’.

Note 1: PIC12F1571/2 only.

2: PIC12(L)F1572 only.

3: Unimplemented, read as ‘`1`’.

TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other Resets
Bank 27											
D8Ch	—	Unimplemented								—	—
D8Dh	—	Unimplemented								—	—
D8Eh	PWMEN	—	—	—	—	—	PWM3EN_A	PWM2EN_A	PWM1EN_A	`---- -000`	`---- -000`
D8Fh	PWMLD	—	—	—	—	—	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A	`---- -000`	`---- -000`
D90h	PWMOUT	—	—	—	—	—	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A	`---- -000`	`---- -000`
D91h	PWM1PHL	PH<7:0>								`xxxx xxxx`	`uuuu uuuu`
D92h	PWM1PHH	PH<15:8>								`xxxx xxxx`	`uuuu uuuu`
D93h	PWM1DCL	DC<7:0>								`xxxx xxxx`	`uuuu uuuu`
D94h	PWM1DCH	DC<15:8>								`xxxx xxxx`	`uuuu uuuu`
D95h	PWM1PRL	PR<7:0>								`xxxx xxxx`	`uuuu uuuu`
D96h	PWM1PRH	PR<15:8>								`xxxx xxxx`	`uuuu uuuu`
D97h	PWM1OFL	OF<7:0>								`xxxx xxxx`	`uuuu uuuu`
D98h	PWM1OFH	OF<15:8>								`xxxx xxxx`	`uuuu uuuu`
D99h	PWM1TMRL	TMR<7:0>								`xxxx xxxx`	`uuuu uuuu`
D9Ah	PWM1TMRH	TMR<15:8>								`xxxx xxxx`	`uuuu uuuu`
D9Bh	PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	PWM1MODE<1:0>		—	—	`0000 00--`	`0000 00--`
D9Ch	PWM1INTE	—	—	—	—	PWM1OFIE	PWM1PHIE	PWM1DCIE	PWM1PRIE	`---- 000`	`---- 000`
D9Dh	PWM1INTF	—	—	—	—	PWM1OFIF	PWM1PHIF	PWM1DCIF	PWM1PRIF	`---- 000`	`---- 000`
D9Eh	PWM1CLKCON	—	PWM1PS<2:0>			—	—	PWM1CS<1:0>		`-000 -000`	`-000 --00`
D9Fh	PWM1LDCON	PWM1LDA	PWM1LDT	—	—	—	—	PWM1LDS<1:0>		`00-- -000`	`00-- --00`
DA0h	PWM1OFCON	—	PWM1OFM<1:0>		PWM1OFO	—	—	PWM1OFS<1:0>		`-000 -000`	`-000 --00`
DA1h	PWM2PHL	PH<7:0>								`xxxx xxxx`	`uuuu uuuu`
DA2h	PWM2PHH	PH<15:8>								`xxxx xxxx`	`uuuu uuuu`
DA3h	PWM2DCL	DC<7:0>								`xxxx xxxx`	`uuuu uuuu`
DA4h	PWM2DCH	DC<15:8>								`xxxx xxxx`	`uuuu uuuu`
DA5h	PWM2PRL	PR<7:0>								`xxxx xxxx`	`uuuu uuuu`
DA6h	PWM2PRH	PR<15:8>								`xxxx xxxx`	`uuuu uuuu`
DA7h	PWM2OFL	OF<7:0>								`xxxx xxxx`	`uuuu uuuu`
DA8h	PWM2OFH	OF<15:8>								`xxxx xxxx`	`uuuu uuuu`
DA9h	PWM2TMRL	TMR<7:0>								`xxxx xxxx`	`uuuu uuuu`
DAAh	PWM2TMRH	TMR<15:8>								`xxxx xxxx`	`uuuu uuuu`
DABh	PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	PWM2MODE<1:0>		—	—	`0000 00--`	`0000 00--`
DACh	PWM2INTE	—	—	—	—	PWM2OFIE	PWM2PHIE	PWM2DCIE	PWM2PRIE	`---- 000`	`---- 000`
DADh	PWM2INTF	—	—	—	—	PWM2OFIF	PWM2PHIF	PWM2DCIF	PWM2PRIF	`---- 000`	`---- 000`
DAEh	PWM2CLKCON	—	PWM2PS<2:0>			—	—	PWM2CS<1:0>		`-000 -000`	`-000 --00`
DAFh	PWM2LDCON	PWM2LDA	PWM2LDT	—	—	—	—	PWM2LDS<1:0>		`00-- -000`	`00-- --00`

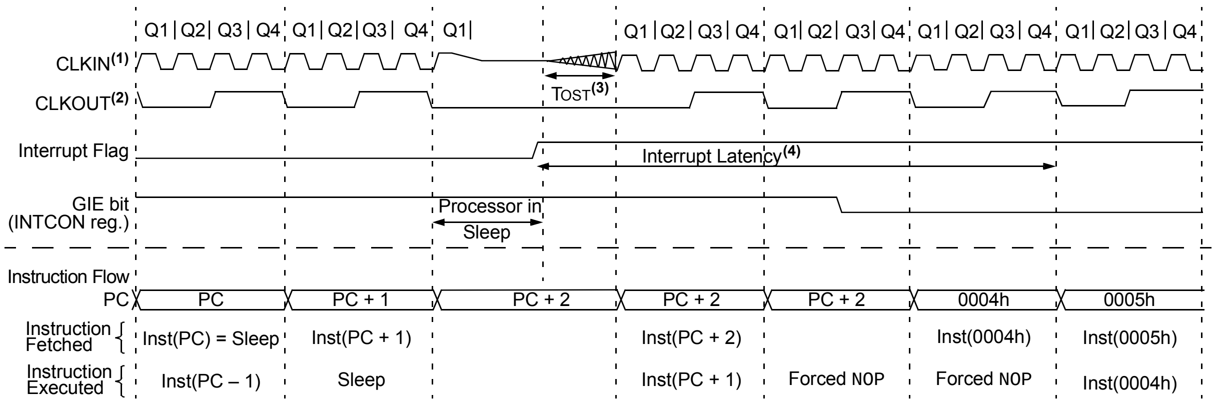
Legend: x = unknown; u = unchanged; q = value depends on condition; — = unimplemented; r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: PIC12F1571/2 only.

2: PIC12(L)F1572 only.

3: Unimplemented, read as '1'.

FIGURE 8-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: External Clock. High, Medium, Low mode assumed.
2: CLKOUT is shown here for timing reference.
3: TOST = 1024 TOSC. This delay does not apply to EC, RC and INTOSC Oscillator modes or Two-Speed Start-up (if available).
4: GIE = 1 assumed. In this case, after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

8.2 Low-Power Sleep Mode

This device contains an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode.

Low-Power Sleep mode allows the user to optimize the operating current in Sleep. Low-Power Sleep mode can be selected by setting the VREGPM bit of the VREGCON register, which puts the LDO and reference circuitry in a low-power state whenever the device is in Sleep.

8.2.1 SLEEP CURRENT VS. WAKE-UP TIME

In the default operating mode, the LDO and reference circuitry remain in the normal configuration while in Sleep. The device is able to exit Sleep mode quickly since all circuits remain active. In Low-Power Sleep mode, when waking up from Sleep, an extra delay time is required for these circuits to return to the normal configuration and stabilize.

The Low-Power Sleep mode is beneficial for applications that stay in Sleep mode for long periods of time. The normal mode is beneficial for applications that need to wake from Sleep quickly and frequently.

8.2.2 PERIPHERAL USAGE IN SLEEP

Some peripherals that can operate in Sleep mode will not operate properly with the Low-Power Sleep mode selected. The LDO will remain in the normal power mode when those peripherals are enabled. The Low-Power Sleep mode is intended for use with these peripherals:

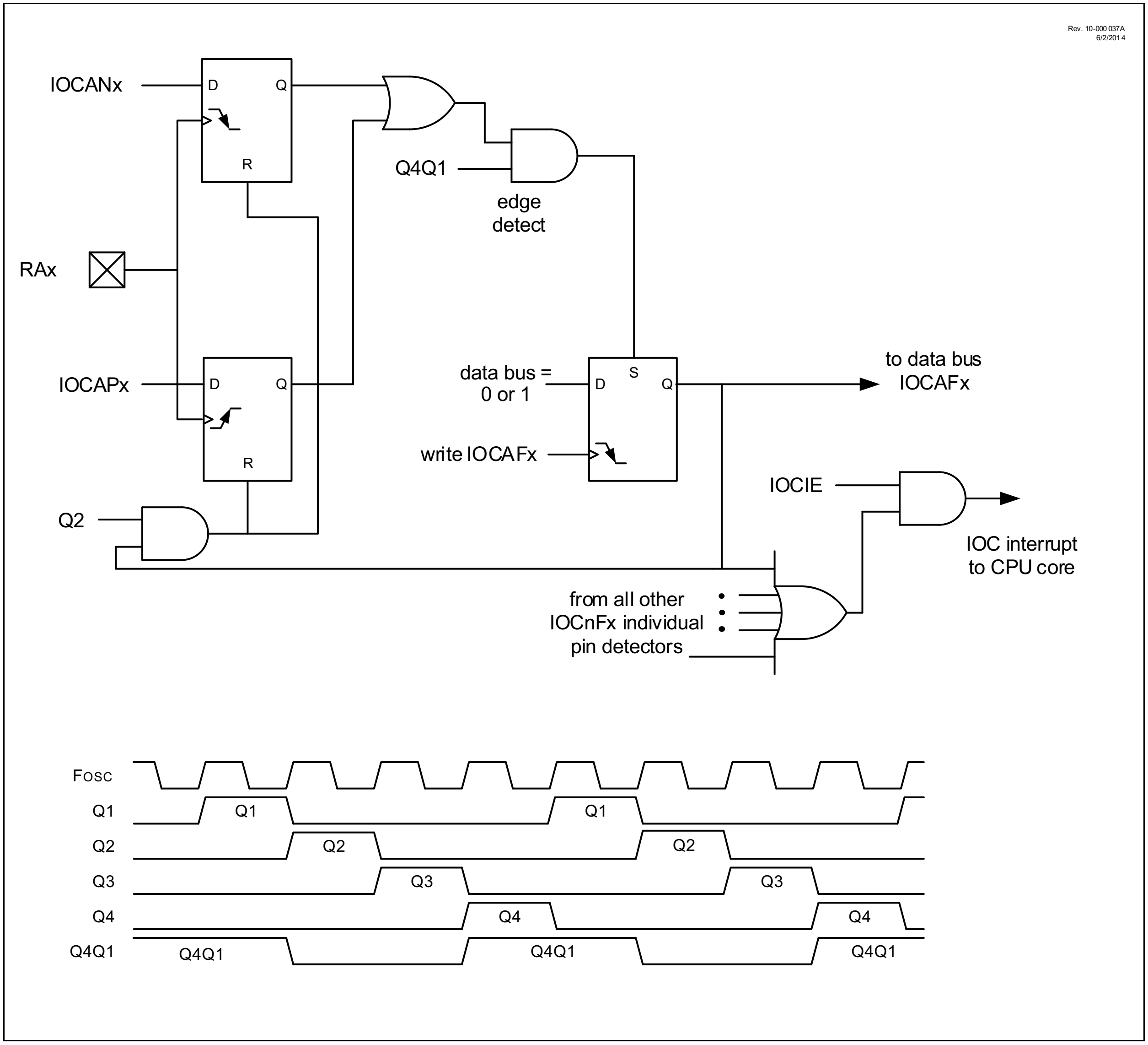
- Brown-out Reset (BOR)
- Watchdog Timer (WDT)
- External interrupt pin/Interrupt-On-Change pins
- Timer1 (with external clock source)

The Complementary Waveform Generator (CWG) module can utilize the HFINTOSC oscillator as either a clock source or as an input source. Under certain conditions, when the HFINTOSC is selected for use with the CWG module, the HFINTOSC will remain active during Sleep. This will have a direct effect on the Sleep mode current.

Please refer to section **Section 23.10 “Operation During Sleep”** for more information.

Note:	The PIC12LF1571/2 does not have a configurable Low-Power Sleep mode. PIC12LF1571/2 is an unregulated device and is always in the lowest power state when in Sleep with no wake-up time penalty. This device has a lower maximum VDD and I/O voltage than the PIC12F1571/2. See **Section 26.0 “Electrical Specifications”** for more information.

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



16.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The positive input source (VSOURCE+) of the DAC can be connected to the:

- External VREF+ pin
- VDD supply voltage
- FVR buffered output

The negative input source (VSOURCE-) of the DAC can be connected to the:

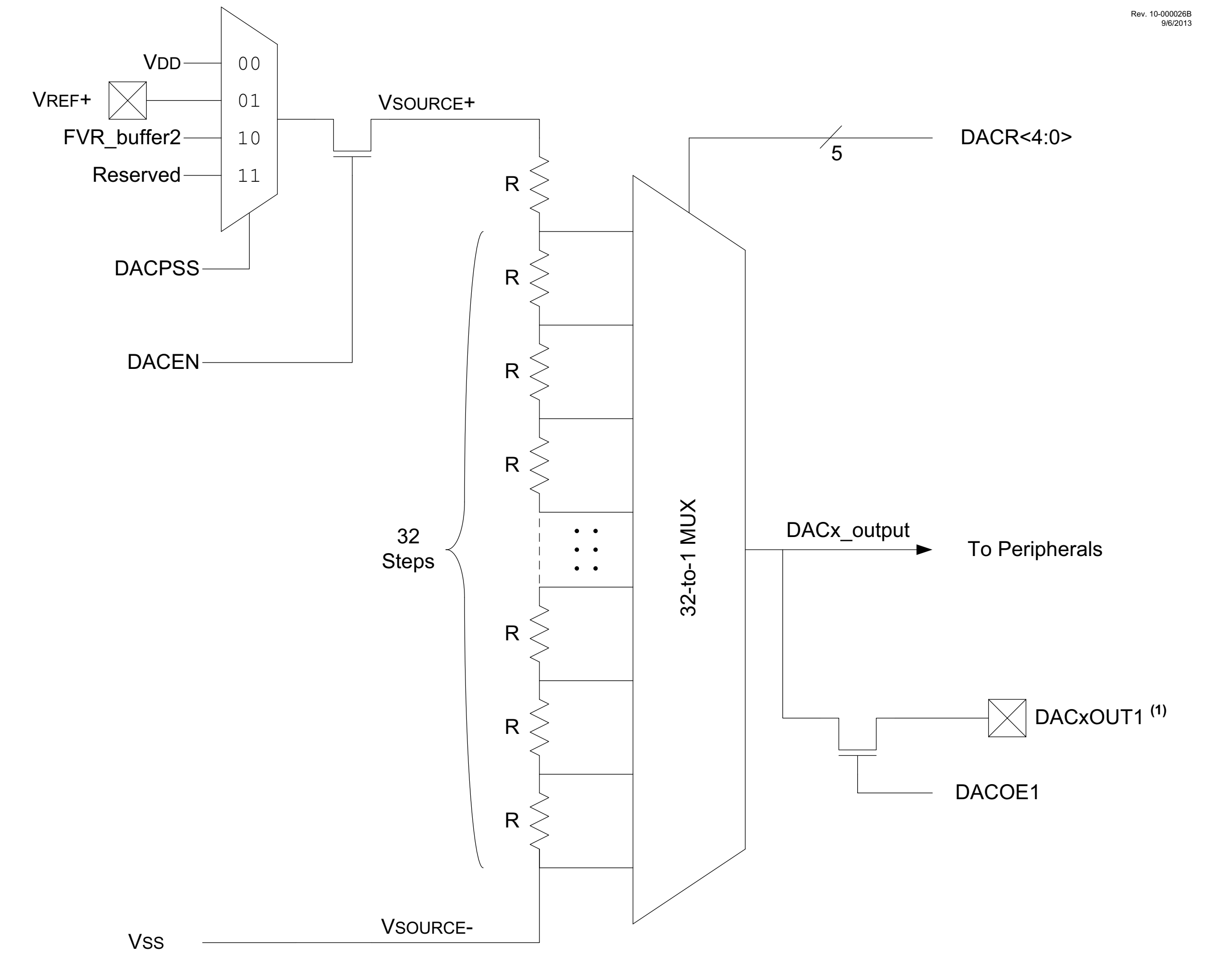
- Vss

The output of the DAC (DACx_output) can be selected as a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACxOUT1 pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the DACEN bit of the DACxCON0 register.

FIGURE 16-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



Note 1: The unbuffered DACx_output is provided on the DACxOUT pin(s).

FIGURE 19-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE

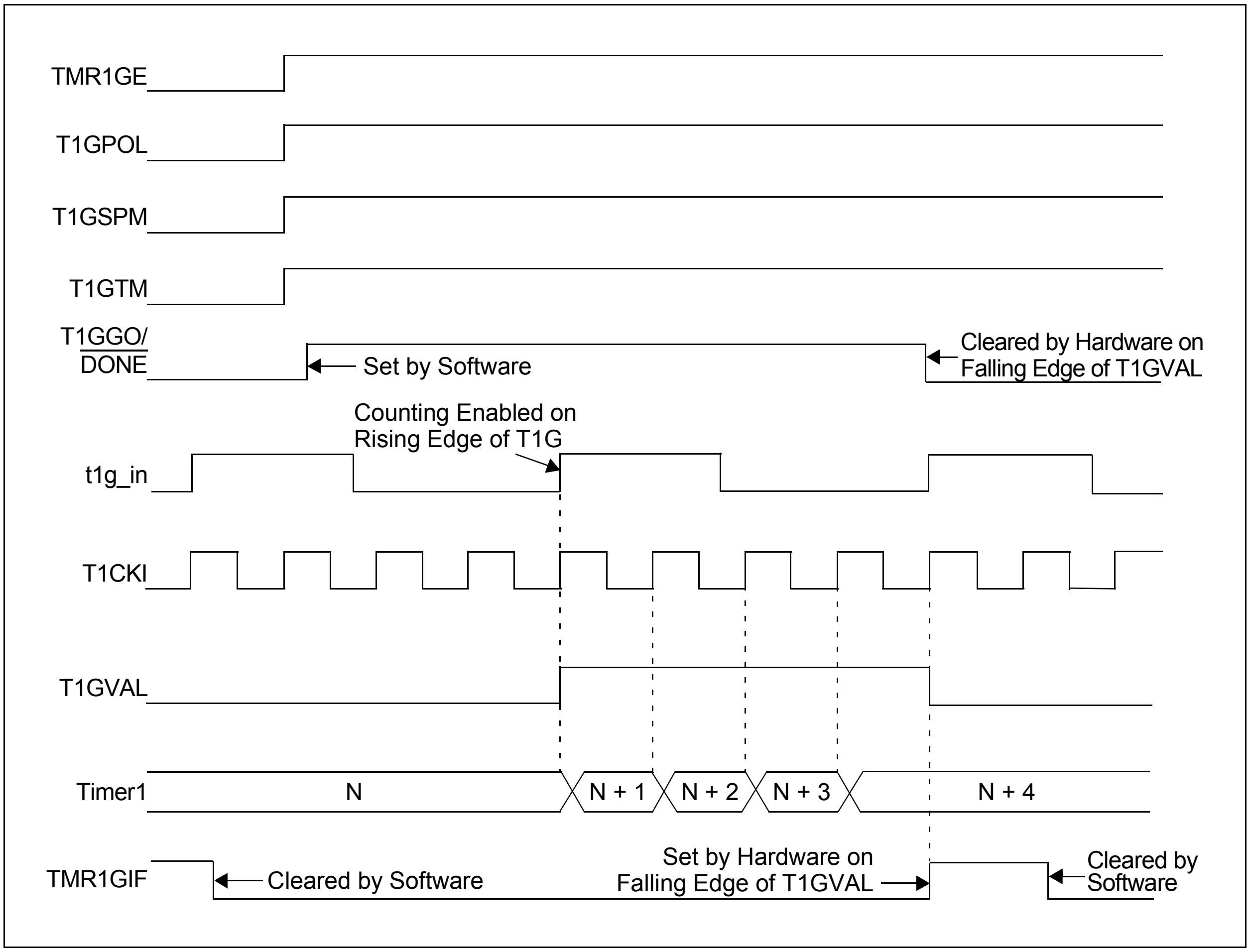


TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	186
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
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	185*
SPBRGL	BRG<7:0>								187*
SPBRGH	BRG<15:8>								187*
TXREG	EUSART Transmit Data Register								177
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	184

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for asynchronous transmission.
* Page provides register information.

Note 1: PIC12(L)F1572 only.

TABLE 21-5: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1202	0.16	207	1200	0.00	143
2400	2404	0.16	129	2400	0.00	119	2404	0.16	103	2400	0.00	71
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17
10417	10417	0.00	29	10286	-1.26	27	10417	0.00	23	10165	-2.42	16
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8
57.6k	—	—	—	57.60k	0.00	7	—	—	—	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)	Actual Rate	% Error	SPBRG Value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5

21.4.2 AUTO-BAUD OVERFLOW

During the course of Automatic Baud Detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRGL register pair. The overflow condition will set the RCIF flag. The counter continues to count until the fifth rising edge is detected on the RX pin. The RCIDL bit will remain false ('0') until the fifth rising edge, at which time, the RDICL bit will set. If the RCREG is read after the overflow occurs, but before the fifth rising edge, the fifth rising edge will set the RCIF again.

Terminating the auto-baud process early to clear an overflow condition will prevent proper detection of the Sync character fifth rising edge. If any falling edges of the Sync character have not yet occurred when the ABDEN bit is cleared, then those will be falsely detected as Start bits. The following steps are recommended to clear the overflow condition:

1. Read RCREG to clear RCIF.
2. If RCIDL is zero, then wait for RCIF and repeat Step 1.
3. Clear the ABDOVF bit.

21.4.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The auto-wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The auto-wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 21-7), and asynchronously if the device is in Sleep mode (Figure 21-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

21.4.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled, the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be ten or more bit times; 13-bit times are recommended for LIN bus or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

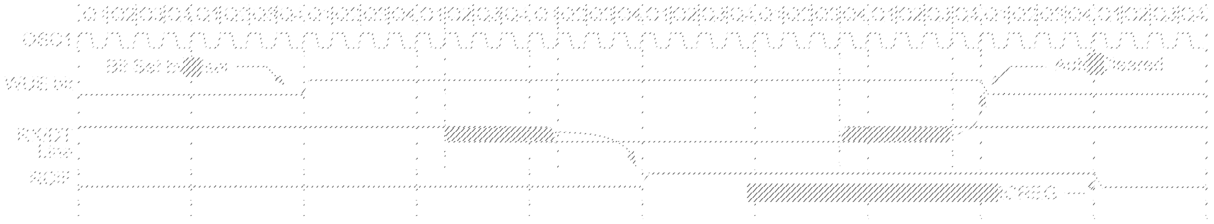
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

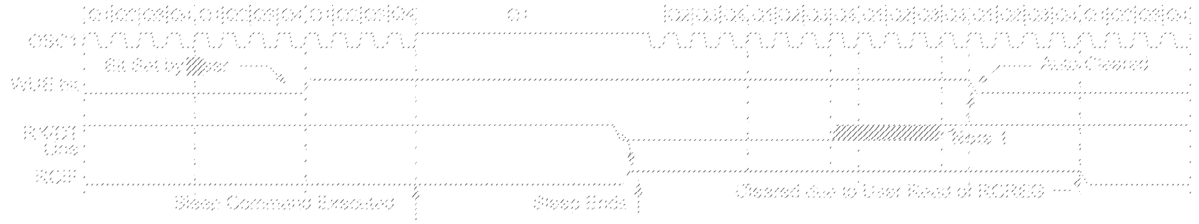
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 21-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION



Note 1: The EUSART remains in Idle while the WUE bit is set.

FIGURE 21-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



Note 1: If the wake-up event requires long oscillator warm-up time, the automatic clearing of the WUE bit can occur while the stposc signal is still active. This sequence should not depend on the presence of Q clocks.

2: The EUSART remains in Idle while the WUE bit is set.

24.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process, allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}$/VPP
- VDD
- VSS

In Program/Verify mode, the program memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™, refer to the *"PIC12(L)F1501/PIC16(L)F150X Memory Programming Specification"* (DS41573).

24.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low, then raising the voltage on $\overline{\text{MCLR}}$/VPP to VIHH.

24.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC® MCUs (Flash) to be programmed using VDD only, without high voltage. When the LVP bit of the Configuration Words is set to '1', the ICSP Low-Voltage Programming Entry mode is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to '0'.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

1. $\overline{\text{MCLR}}$ is brought to VIL.
2. A 32-bit key sequence is presented on ICSPDAT while clocking ICSPCLK.

Once the key sequence is complete, $\overline{\text{MCLR}}$ must be held at VIL for as long as Program/Verify mode is to be maintained.

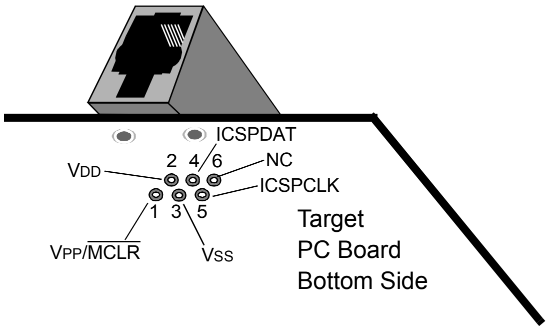
If Low-Voltage Programming is enabled (LVP = 1), the $\overline{\text{MCLR}}$ Reset function is automatically enabled and cannot be disabled. See **Section 6.5 "MCLR"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

24.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP™ header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6-connector) configuration. See Figure 24-1.

FIGURE 24-1: ICD RJ-11 STYLE CONNECTOR INTERFACE



Pin Description*

1 = VPP/$\overline{\text{MCLR}}$

2 = VDD Target

3 = VSS (ground)

4 = ICSPDAT

5 = ICSPCLK

6 = No Connect

Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 24-2.

25.2 Instruction Descriptions

ADDFSR	Add Literal to FSRn
Syntax:	[*label*] ADDFSR FSRn, k
Operands:	$-32 \le k \le 31$ $n \in [0, 1]$
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal ‘k’ is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wraparound.

ADDLW	Add literal and W
Syntax:	[*label*] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal ‘k’ and the result is placed in the W register.

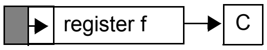
ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ADDWFC	ADD W and CARRY bit to f
Syntax:	[*label*] ADDWFC f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location ‘f’. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed in data memory location ‘f’.

ANDLW	AND literal with W
Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND’ed with the 8-bit literal ‘k’. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ASRF	Arithmetic Right Shift
Syntax:	[*label*] ASRF f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<7>)→ dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.



CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

CALLW	Subroutine Call With W
Syntax:	[*label*] CALLW
Operands:	None
Operation:	(PC) +1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
Status Affected:	None
Description:	Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a 2-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

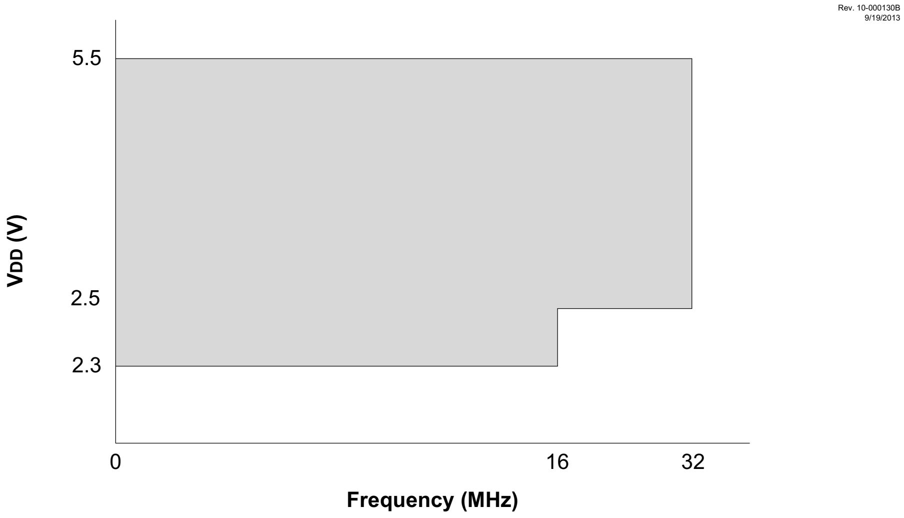
CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(\overline{f})$ → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

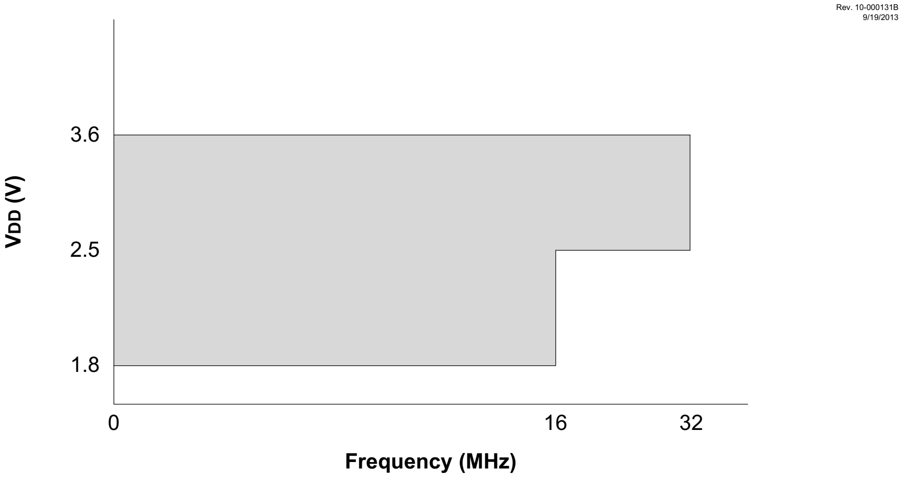
FIGURE 26-1: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC12F1571/2 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: Refer to Table 26-7 for each Oscillator mode’s supported frequencies.

FIGURE 26-2: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC12LF1571/2 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: Refer to Table 26-7 for each Oscillator mode’s supported frequencies.

FIGURE 27-11: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC12F1571/2 ONLY

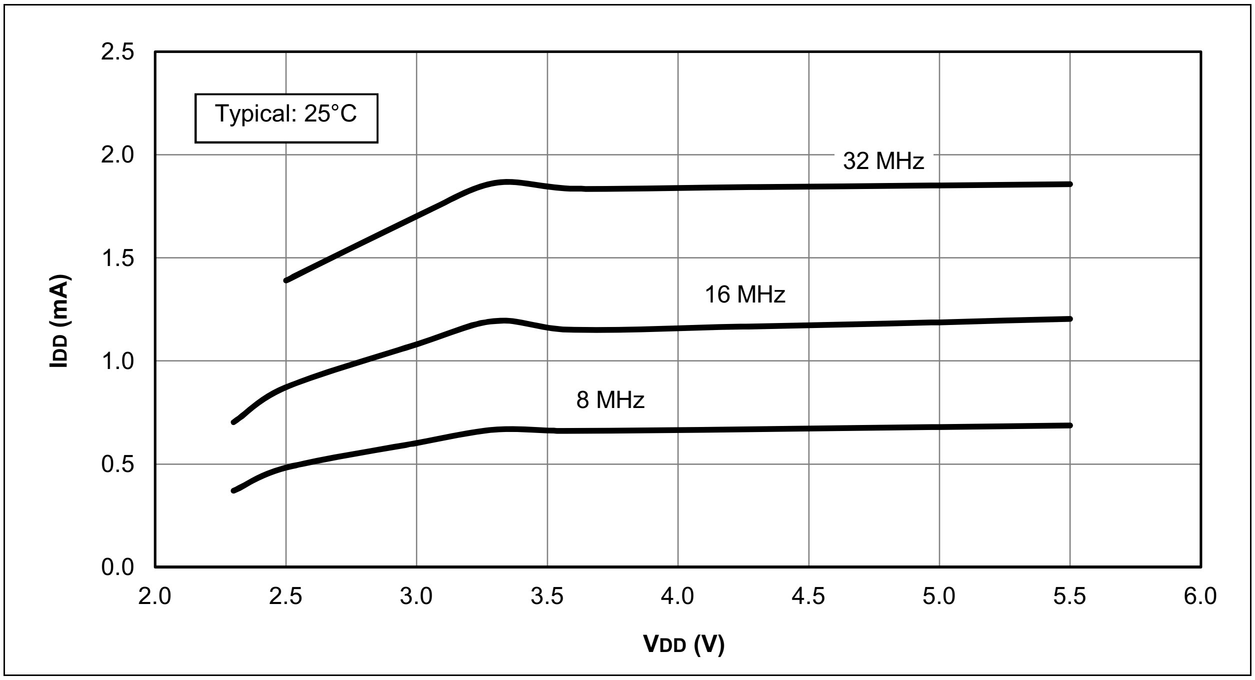


FIGURE 27-12: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC12F1571/2 ONLY

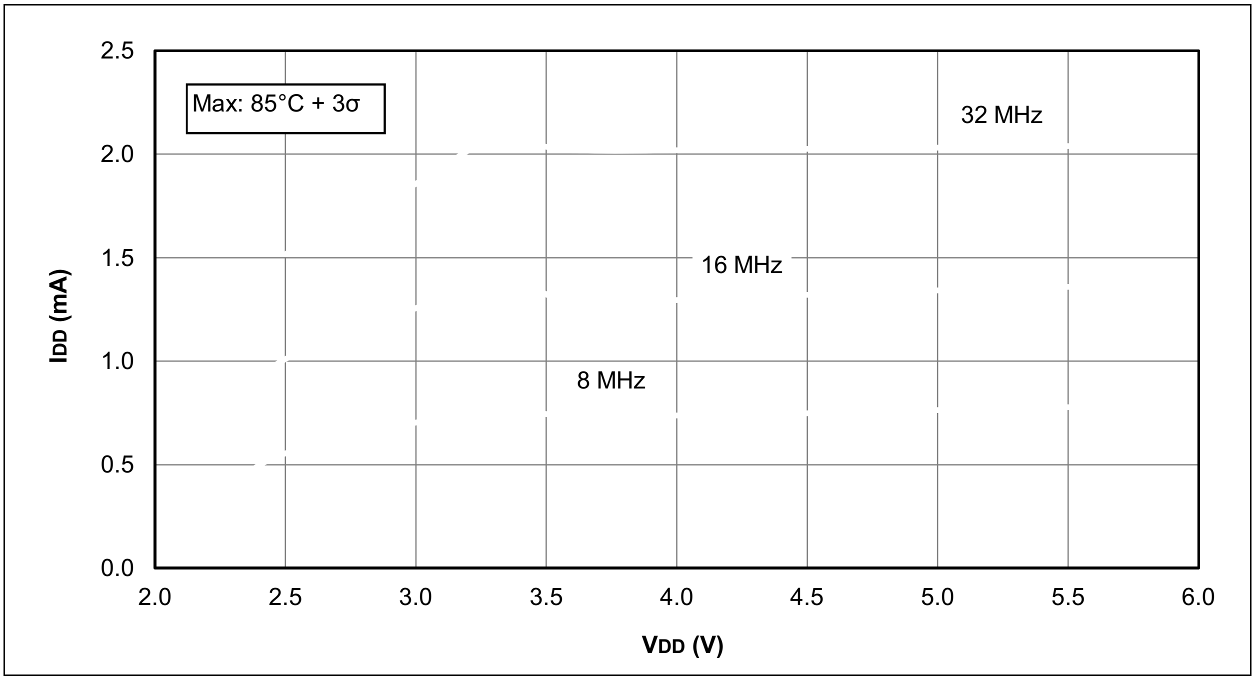


FIGURE 27-27: IPD, BROWN-OUT RESET (BOR), BORV = 1, PIC12LF1571/2 ONLY

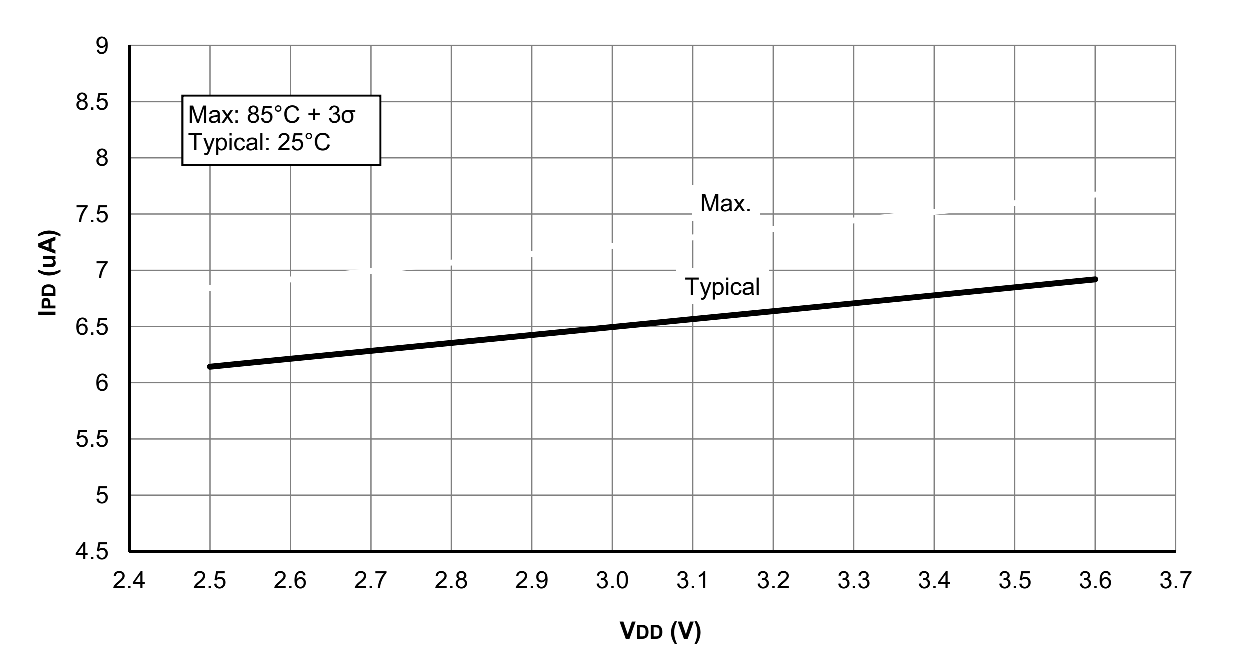


FIGURE 27-28: IPD, BROWN-OUT RESET (BOR), BORV = 1, PIC12F1571/2 ONLY

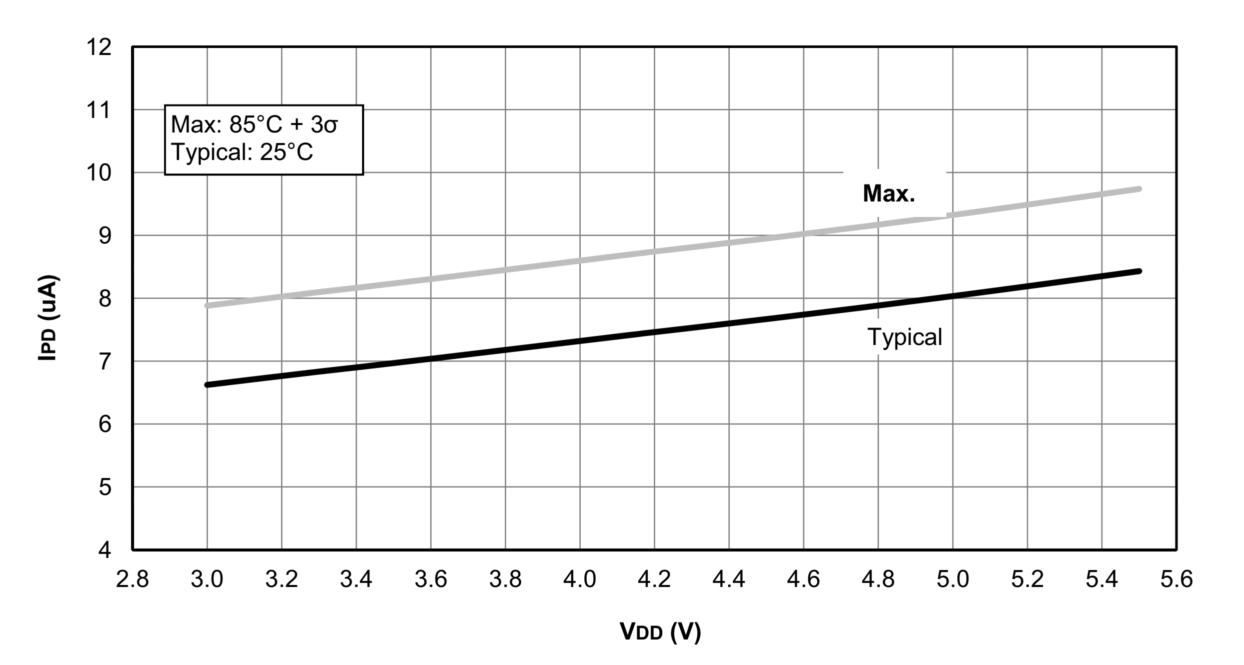


TABLE 29-1: 8-LEAD 3x3x0.9 DFN (MF) TOP MARKING

Part Number	Marking
PIC12F1571-E/MF	MFY0/YYWW/NNN
PIC12F1572-E/MF	MGA0/YYWW/NNN
PIC12F1571-I/MF	MFZ0
PIC12F1572-I/MF	MGB0
PIC12LF1571-E/MF	MGC0
PIC12LF1572-E/MF	MGE0
PIC12LF1571-I/MF	MGD0
PIC12LF1572-I/MF	MGF0

TABLE 29-2: 8-LEAD 3x3x0.5 UDFN (RF) TOP MARKING

Part Number	Marking
PIC12F1571-E/MF	MFY0/YYWW/NNN
PIC12F1572-E/MF	MGA0/YYWW/NNN
PIC12F1571-I/MF	MFZ0
PIC12F1572-I/MF	MGB0
PIC12LF1571-E/MF	MGC0
PIC12LF1572-E/MF	MGE0
PIC12LF1571-I/MF	MGD0
PIC12LF1572-I/MF	MGF0

NOTES: