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	LINbus, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1578-i-so

TABLE 3-9: PIC16(L)F1574/5/8/9 MEMORY MAP, BANKS 16-23

BANK16		BANK17		BANK18		BANK19		BANK20		BANK21		BANK22		BANK23	
800h	Core Registers (Table 3-2)	880h	Core Registers (Table 3-2)	900h	Core Registers (Table 3-2)	980h	Core Registers (Table 3-2)	A00h	Core Registers (Table 3-2)	A80h	Core Registers (Table 3-2)	B00h	Core Registers (Table 3-2)	B80h	Core Registers (Table 3-2)
80Bh		88Bh		90Bh		98Bh		A0Bh		A8Bh		B0Bh		B8Bh	
80Ch	—	88Ch	—	90Ch	—	98Ch	—	A0Ch	—	A8Ch	—	B0Ch	—	B8Ch	—
80Dh	—	88Dh	—	90Dh	—	98Dh	—	A0Dh	—	A8Dh	—	B0Dh	—	B8Dh	—
80Eh	—	88Eh	—	90Eh	—	98Eh	—	A0Eh	—	A8Eh	—	B0Eh	—	B8Eh	—
80Fh	—	88Fh	—	90Fh	—	98Fh	—	A0Fh	—	A8Fh	—	B0Fh	—	B8Fh	—
810h	—	890h	—	910h	—	990h	—	A10h	—	A90h	—	B10h	—	B90h	—
811h	—	891h	—	911h	—	991h	—	A11h	—	A91h	—	B11h	—	B91h	—
812h	—	892h	—	912h	—	992h	—	A12h	—	A92h	—	B12h	—	B92h	—
813h	—	893h	—	913h	—	993h	—	A13h	—	A93h	—	B13h	—	B93h	—
814h	—	894h	—	914h	—	994h	—	A14h	—	A94h	—	B14h	—	B94h	—
815h	—	895h	—	915h	—	995h	—	A15h	—	A95h	—	B15h	—	B95h	—
816h	—	896h	—	916h	—	996h	—	A16h	—	A96h	—	B16h	—	B96h	—
817h	—	897h	—	917h	—	997h	—	A17h	—	A97h	—	B17h	—	B97h	—
818h	—	898h	—	918h	—	998h	—	A18h	—	A98h	—	B18h	—	B98h	—
819h	—	899h	—	919h	—	999h	—	A19h	—	A99h	—	B19h	—	B99h	—
81Ah	—	89Ah	—	91Ah	—	99Ah	—	A1Ah	—	A9Ah	—	B1Ah	—	B9Ah	—
81Bh	—	89Bh	—	91Bh	—	99Bh	—	A1Bh	—	A9Bh	—	B1Bh	—	B9Bh	—
81Ch	—	89Ch	—	91Ch	—	99Ch	—	A1Ch	—	A9Ch	—	B1Ch	—	B9Ch	—
81Dh	—	89Dh	—	91Dh	—	99Dh	—	A1Dh	—	A9Dh	—	B1Dh	—	B9Dh	—
81Eh	—	89Eh	—	91Eh	—	99Eh	—	A1Eh	—	A9Eh	—	B1Eh	—	B9Eh	—
81Fh	—	89Fh	—	91Fh	—	99Fh	—	A1Fh	—	A9Fh	—	B1Fh	—	B9Fh	—
820h	Unimplemented Read as '0'	8A0h	Unimplemented Read as '0'	920h	Unimplemented Read as '0'	9A0h	Unimplemented Read as '0'	A20h	Unimplemented Read as '0'	AA0h	Unimplemented Read as '0'	B20h	Unimplemented Read as '0'	BA0h	Unimplemented Read as '0'
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh		BEFh	
870h	Accesses 70h – 7Fh	8F0h	Accesses 70h – 7Fh	970h	Accesses 70h – 7Fh	9F0h	Accesses 70h – 7Fh	A70h	Accesses 70h – 7Fh	AF0h	Accesses 70h – 7Fh	B70h	Accesses 70h – 7Fh	BF0h	Accesses 70h – 7Fh
87Fh		8FFh		97Fh		9FFh		A7Fh		AFFh		B7Fh		BFFh	

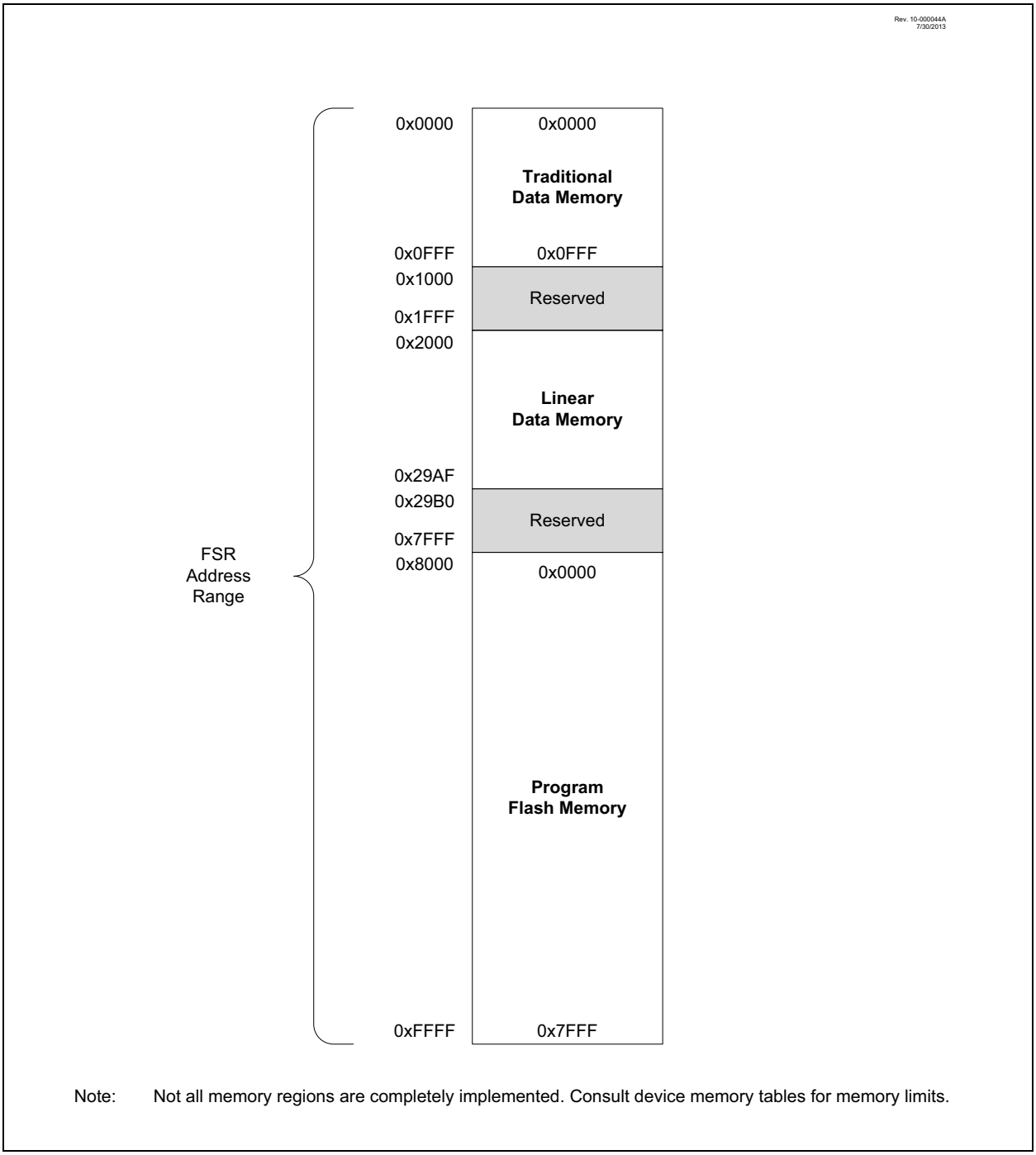
Legend: = Unimplemented data memory locations, read as '0'.

TABLE 3-10: PIC16(L)F1574/5/8/9 MEMORY MAP, BANKS 24-31

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31	
C00h	Core Registers (Table 3-2)	C80h	Core Registers (Table 3-2)	D00h	Core Registers (Table 3-2)	D80h	Core Registers (Table 3-2)	E00h	Core Registers (Table 3-2)	E80h	Core Registers (Table 3-2)	F00h	Core Registers (Table 3-2)	F80h	Core Registers (Table 3-2)
C0Bh		C8Bh		D0Bh		D8Bh		E0Bh		E8Bh		F0Bh		F8Bh	
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	See Table 3-11	E0Ch	See Table 3-12	E8Ch	See Table 3-12	F0Ch	—	F8Ch	See Table 3-13
C0Dh	—	C8Dh	—	D0Dh	—							F0Dh	—		
C0Eh	—	C8Eh	—	D0Eh	—							F0Eh	—		
C0Fh	—	C8Fh	—	D0Fh	—							F0Fh	—		
C10h	—	C90h	—	D10h	—							F10h	—		
C11h	—	C91h	—	D11h	—							F11h	—		
C12h	—	C92h	—	D12h	—							F12h	—		
C13h	—	C93h	—	D13h	—							F13h	—		
C14h	—	C94h	—	D14h	—							F14h	—		
C15h	—	C95h	—	D15h	—							F15h	—		
C16h	—	C96h	—	D16h	—							F16h	—		
C17h	—	C97h	—	D17h	—							F17h	—		
C18h	—	C98h	—	D18h	—							F18h	—		
C19h	—	C99h	—	D19h	—							F19h	—		
C1Ah	—	C9Ah	—	D1Ah	—							F1Ah	—		
C1Bh	—	C9Bh	—	D1Bh	—							F1Bh	—		
C1Ch	—	C9Ch	—	D1Ch	—							F1Ch	—		
C1Dh	—	C9Dh	—	D1Dh	—							F1Dh	—		
C1Eh	—	C9Eh	—	D1Eh	—							F1Eh	—		
C1Fh	—	C9Fh	—	D1Fh	—							F1Fh	—		
C20h	Unimplemented Read as ‘0’	CA0h	Unimplemented Read as ‘0’	D20h	Unimplemented Read as ‘0’							F20h	Unimplemented Read as ‘0’		
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh		FEFh	
C70h	Accesses 70h – 7Fh	CF0h	Accesses 70h – 7Fh	D70h	Accesses 70h – 7Fh	DF0h	Accesses 70h – 7Fh	E70h	Accesses 70h – 7Fh	EF0h	Accesses 70h – 7Fh	F70h	Accesses 70h – 7Fh	FF0h	Accesses 70h – 7Fh
CFFh		CFFh		D7Fh		DFFh		E7Fh		EFFh		F7Fh		FFFh	

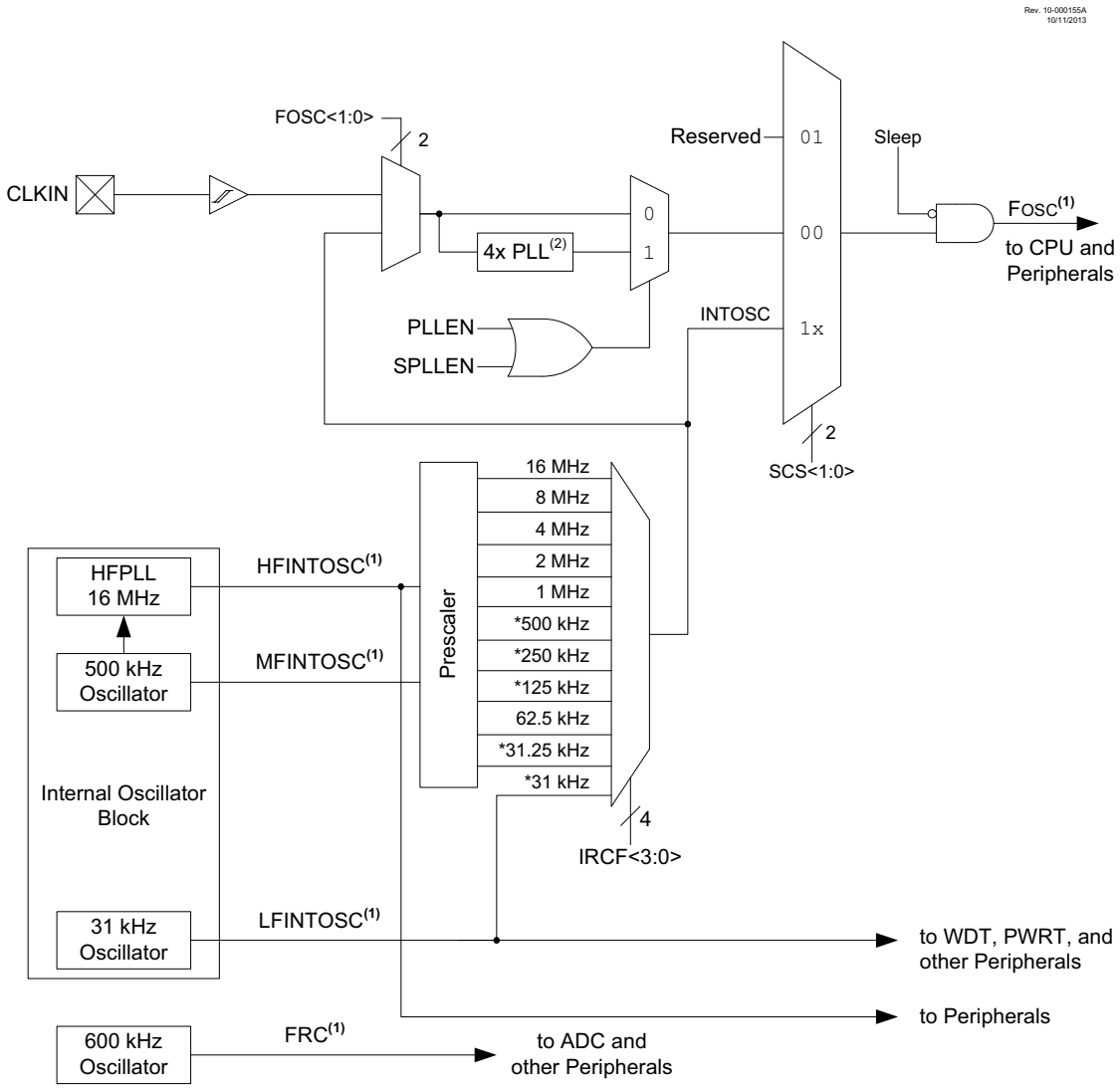
Legend: = Unimplemented data memory locations, read as ‘0’

FIGURE 3-9: INDIRECT ADDRESSING



Note: Not all memory regions are completely implemented. Consult device memory tables for memory limits.

FIGURE 5-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



Note 1: See **Section 5.2 "Clock Source Types"**.

2: If FOSC<1:0> = 00, 4x PLL can only be used if IRCF<3:0> = 1110.

5.2.2.7 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Words must be set to use the INTOSC source as the device system clock (FOSC<1:0> = 00).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<1:0> in Configuration Words (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to the 8 MHz HFINTOSC set to use (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4x PLL, or the PLLEN bit of the Configuration Words must be programmed to a '1'.

Note:	When using the PLLEN bit of the Configuration Words, the 4x PLL cannot be disabled by software and the 8 MHz HFINTOSC option will no longer be available.

The 4x PLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to '1x'. The SCS bits must be set to '00' to use the 4x PLL with the internal oscillator.

5.2.2.8 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-3). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

See Figure 5-3 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 5-1.

Start-up delay specifications are located in the oscillator tables of **Section 27.0 "Electrical Specifications"**.

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		69
OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	70
OSCTUNE	—	—	TUN<5:0>						71

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

TABLE 5-3: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		

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

7.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

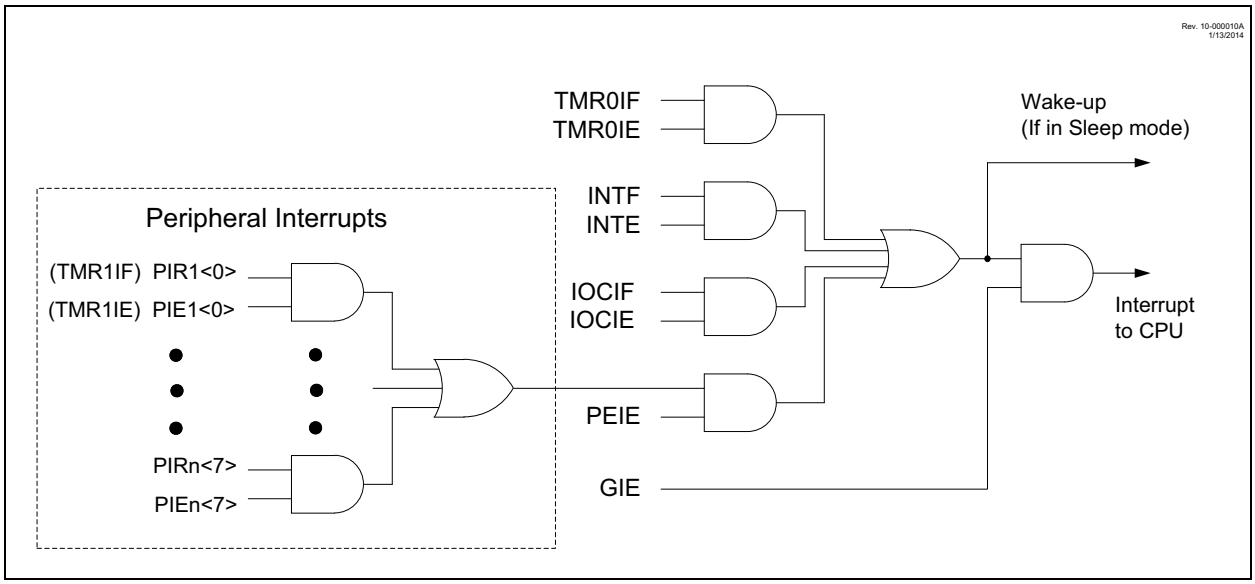
This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Automatic Context Saving

Many peripherals produce interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 7-1.

FIGURE 7-1: INTERRUPT LOGIC



17.6 Register Definitions: DAC Control

REGISTER 17-1: DACCON0: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
DACEN	—	DACOE	—	DACPSS<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 **DACEN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DAC voltage level is output on the DACOUT1 pin
0 = DAC voltage level is disconnected from the DACOUT1 pin

bit 4 **Unimplemented:** Read as ‘0’

bit 3-2 **DACPSS<1:0>:** DAC Positive Source Select bits
11 = Reserved
10 = FVR_buffer2
01 = V_{REF+} pin
00 = V_{DD}

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 17-2: DACCON1: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DACR<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **DACR<4:0>:** DAC Voltage Output Select bits

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>		—	—	168
DACCON1	—	—	—	DACR<4:0>					168

Legend: — = Unimplemented location, read as ‘0’. Shaded cells are not used with the DAC module.

18.2.5 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 18-2 shows the output state versus input conditions, including polarity control.

TABLE 18-2: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

18.2.6 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1' which selects the Normal Speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

18.3 Analog Input Connection Considerations

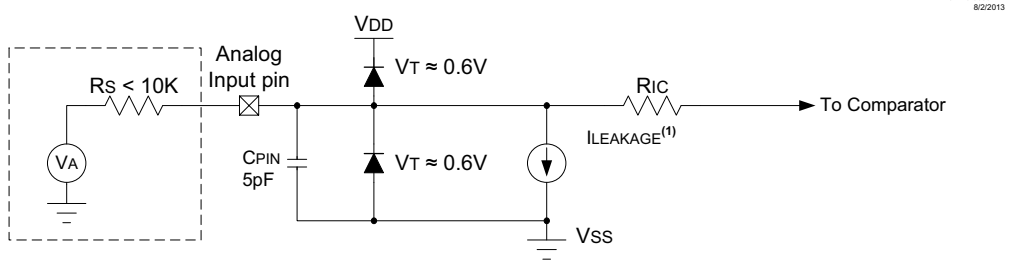
A simplified circuit for an analog input is shown in Figure 18-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward-biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 18-3: ANALOG INPUT MODEL



Legend: CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage
VT = Threshold Voltage

Note 1: See **Section 27.0 "Electrical Specifications"**.

21.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4).

TMR2 increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/ postscaler (see **Section 21.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note:	TMR2 is not cleared when T2CON is written.

21.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (T2_match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE of the PIE1 register.

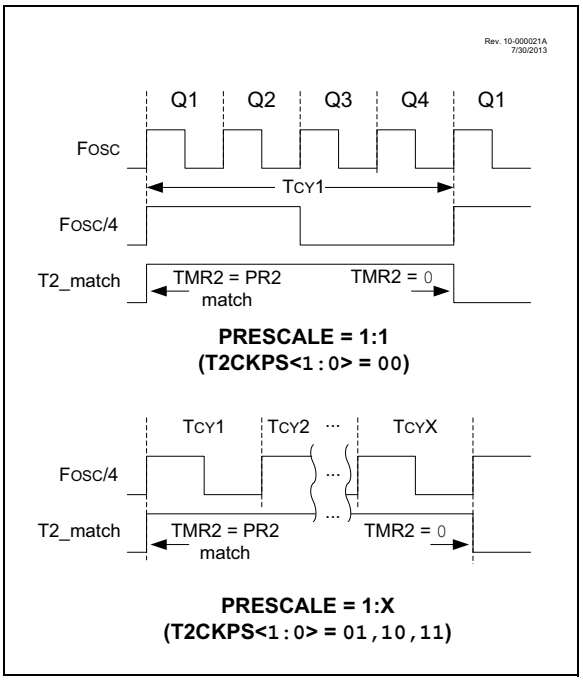
A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS<3:0>, of the T2CON register.

21.3 Timer2 Output

The output of TMR2 is T2_match.

The T2_match signal is synchronous with the system clock. Figure 21-3 shows two examples of the timing of the T2_match signal relative to FOSC and prescale value, T2CKPS<1:0>. The upper diagram illustrates 1:1 prescale timing and the lower diagram, 1:X prescale timing.

FIGURE 21-3: T2_MATCH TIMING DIAGRAM



21.4 Timer2 Operation During Sleep

Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

REGISTER 22-3: BAUDCON: BAUD RATE CONTROL REGISTER

R/W-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care

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

bit 4 **SCKP**: Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Transmit inverted data to the TX/CK pin
0 = Transmit non-inverted data to the TX/CK pin
Synchronous mode:
1 = Data is clocked on rising edge of the clock
0 = Data is clocked on falling edge of the clock

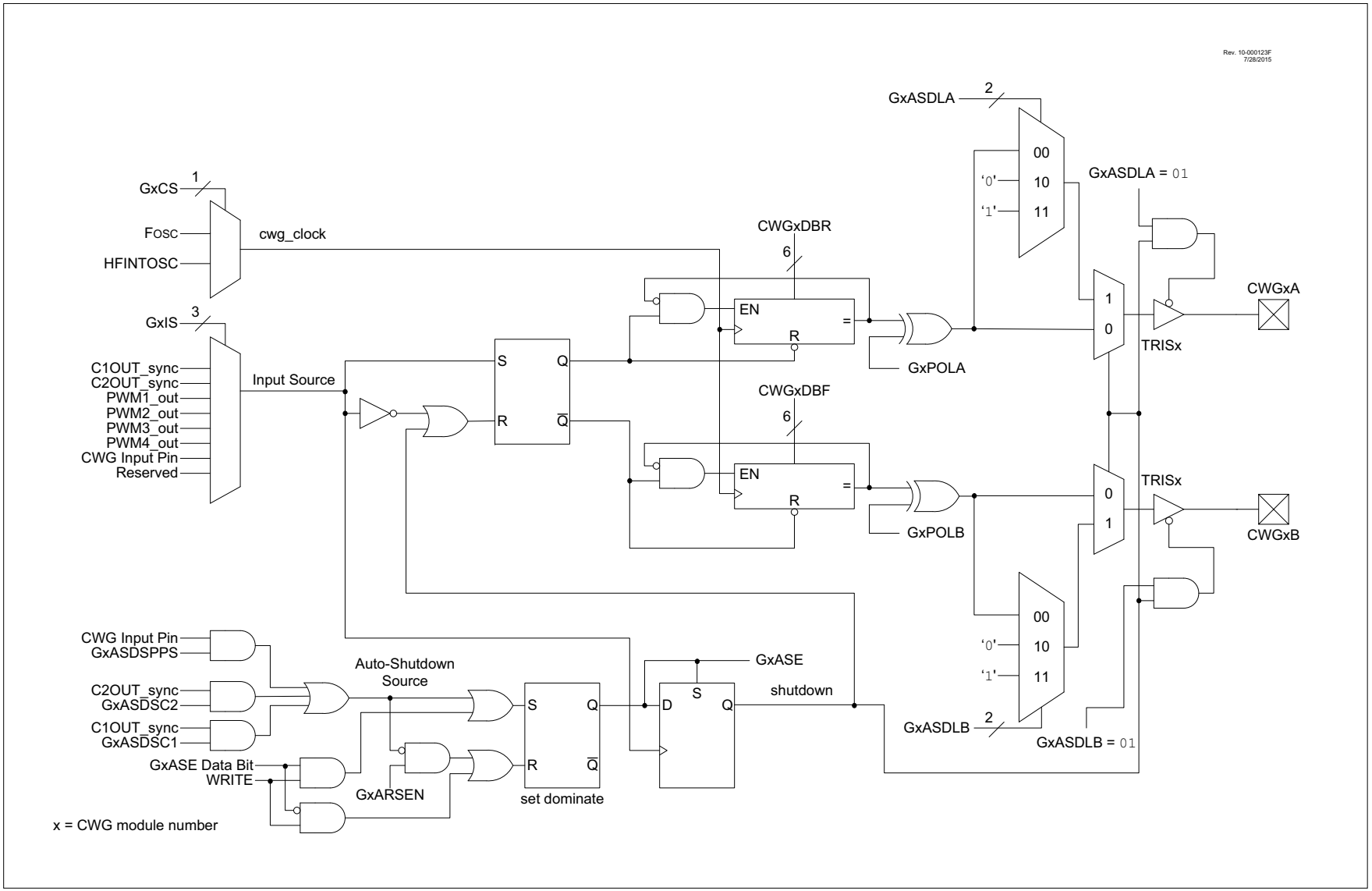
bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

bit 2 **Unimplemented:** Read as '0'

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge. No character will be received, RCIF bit will be set. WUE will automatically clear after RCIF is set.
0 = Receiver is operating normally
Synchronous mode:
Don't care

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don't care

FIGURE 24-1: SIMPLIFIED CWG BLOCK DIAGRAM



24.10 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep, provided that the CWG module is enabled, the input source is active, and the HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, the CPU will go idle during Sleep, but the CWG will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

24.11 Configuring the CWG

The following steps illustrate how to properly configure the CWG to ensure a synchronous start:

1. Ensure that the TRIS control bits corresponding to CWGxA and CWGxB are set so that both are configured as inputs.
2. Clear the GxEN bit, if not already cleared.
3. Set desired dead-band times with the CWGxDBR and CWGxDBF registers.
4. Setup the following controls in CWGxCON2 auto-shutdown register:
   - Select desired shutdown source.
   - Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
   - Set the GxASE bit and clear the GxARSEN bit.
5. Select the desired input source using the CWGxCON1 register.
6. Configure the following controls in CWGxCON0 register:
   - Select desired clock source.
   - Select the desired output polarities.
7. Set the GxEN bit.
8. Clear TRIS control bits corresponding to CWGxA and CWGxB to be used to configure those pins as outputs.
9. If auto-restart is to be used, set the GxARSEN bit and the GxASE bit will be cleared automatically. Otherwise, clear the GxASE bit to start the CWG.

24.11.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the GxASDLA and GxASDLB bits of the CWGxCON1 register (Register 24-3). GxASDLA controls the CWG1A override level and GxASDLB controls the CWG1B override level. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not apply to the override level.

24.11.2 AUTO-SHUTDOWN RESTART

After an auto-shutdown event has occurred, there are two ways to have resume operation:

- Software controlled
- Auto-restart

The restart method is selected with the GxARSEN bit of the CWGxCON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 24-5 and Figure 24-6.

24.11.2.1 Software Controlled Restart

When the GxARSEN bit of the CWGxCON2 register is cleared, the CWG must be restarted after an auto-shutdown event by software.

Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the GxASE bit will remain set. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

24.11.2.2 Auto-Restart

When the GxARSEN bit of the CWGxCON2 register is set, the CWG will restart from the auto-shutdown state automatically.

The GxASE bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

LSLF **Logical Left Shift**

Syntax: [*label*] LSLF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF **Logical Right Shift**

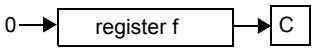
Syntax: [*label*] LSRF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: 0 → 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. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

TABLE 27-3: POWER-DOWN CURRENTS (I_{PD})(1,2)

PIC16LF1574/5/8/9		Operating Conditions: (unless otherwise stated) Low-Power Sleep Mode						
PIC16F1574/5/8/9		Low-Power Sleep Mode, VREGPM = 1						
Param. No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							V_{DD}	Note
D022	Base I_{PD}	—	0.10	1	8	µA	1.8	WDT, BOR, and FVR disabled, all Peripherals inactive
		—	0.10	2	9	µA	3.0	
D022	Base I_{PD}	—	0.3	3	10	µA	2.3	WDT, BOR, and FVR disabled, all Peripherals inactive, Low-Power Sleep mode, VREGPM = 1
		—	0.4	4	12	µA	3.0	
		—	0.5	6	15	µA	5.0	
D022A	Base I_{PD}	—	10.4	16	18	µA	2.3	WDT, BOR, and FVR disabled, all Peripherals inactive, Normal Power Sleep mode, VREGPM = 0
		—	12.7	18	20	µA	3.0	
		—	13.8	21	26	µA	5.0	
D023		—	0.4	2	9	µA	1.8	WDT Current
		—	0.6	3	10	µA	3.0	
D023		—	0.6	6	15	µA	2.3	WDT Current
		—	0.7	7	20	µA	3.0	
		—	0.9	8	22	µA	5.0	
D023A		—	15	28	30	µA	1.8	FVR Current
		—	26	33	34	µA	3.0	
D023A		—	19	28	30	µA	2.3	FVR Current
		—	22	35	36	µA	3.0	
		—	23	38	41	µA	5.0	
D024		—	7.5	17	20	µA	3.0	BOR Current
D024		—	8.1	17	30	µA	3.0	BOR Current
		—	9.2	20	40	µA	5.0	
D24A		—	0.3	4	10	µA	3.0	LPBOR Current
D24A		—	0.5	5	14	µA	3.0	LPBOR Current
		—	0.6	8	17	µA	5.0	
D026		—	0.1	1.5	9	µA	1.8	ADC Current **(Note 3)**, No conversion in progress
		—	0.1	2.7	10	µA	3.0	
D026		—	0.3	4	11	µA	2.3	ADC Current **(Note 3)**, No conversion in progress
		—	0.4	5	13	µA	3.0	
		—	0.5	8	16	µA	5.0	
D026A*		—	288	—	—	µA	1.8	ADC Current **(Note 3)**, Conversion in progress
		—	288	—	—	µA	3.0	
D026A*		—	322	—	—	µA	2.3	ADC Current **(Note 3)**, Conversion in progress
		—	322	—	—	µA	3.0	
		—	322	—	—	µA	5.0	

* 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: The peripheral Δ current can be determined by subtracting the base I_{PD} current from this limit. Max. values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{SS}.

3: ADC clock source is FRC.

FIGURE 27-11: ADC CONVERSION TIMING (ADC CLOCK Fosc-BASED)

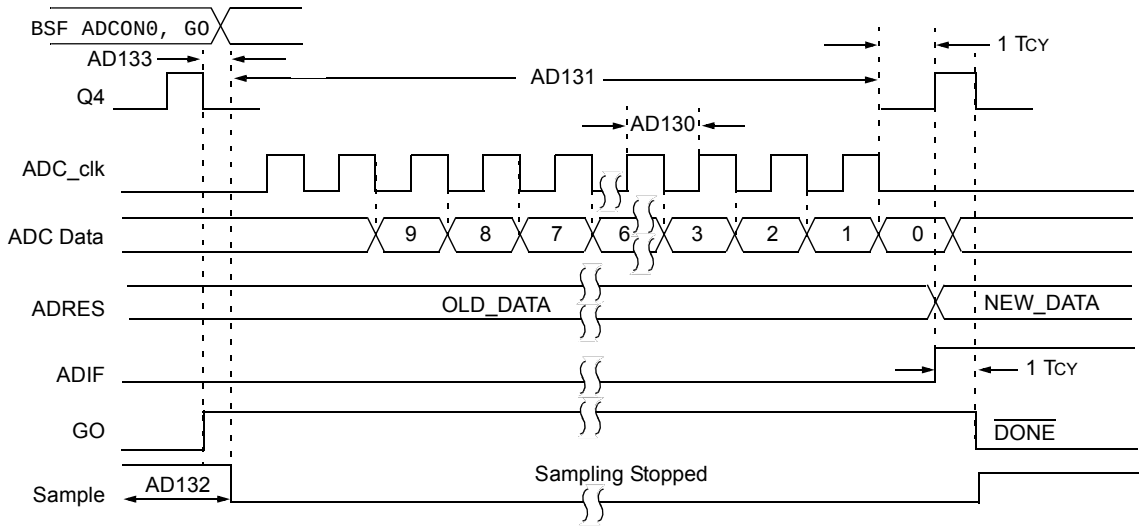
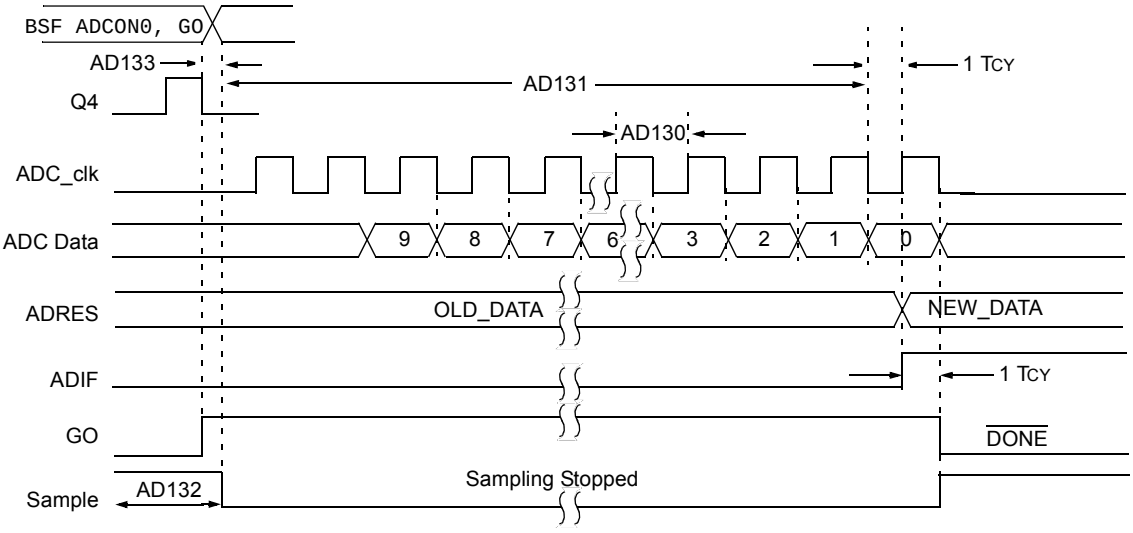


FIGURE 27-12: ADC CONVERSION TIMING (ADC CLOCK FROM FRC)



Note 1: If the ADC clock source is selected as FRC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.

TABLE 27-14: ADC CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD130*	TAD	ADC Clock Period (TADC)	1.0	—	6.0	µs	FOSC-based
		ADC Internal FRC Oscillator Period (TFRC)	1.0	2.0	6.0	µs	ADCS<2:0> = x11 (ADC FRC mode)
AD131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	11	—	TAD	Set GO/$\overline{\text{DONE}}$ bit to conversion complete
AD132*	TACQ	Acquisition Time	—	5.0	—	µs	
AD133*	THCD	Holding Capacitor Disconnect Time	—	1/2 TAD	—		FOSC-based
			—	1/2 TAD + 1TCY	—		ADCS<2:0> = x11 (ADC FRC mode)

* 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: The ADRES register may be read on the following TCY cycle.

TABLE 27-15: COMPARATOR SPECIFICATIONS(1)

Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	**Sym.**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
CM01	VIOFF	Input Offset Voltage	—	±7.5	±60	mV	CxSP = 1, VICM = VDD/2
CM02	VICM	Input Common Mode Voltage	0	—	VDD	V	
CM03	CMRR	Common Mode Rejection Ration	—	50	—	dB	
CM04A	TRESP(2)	Response Time Rising Edge	—	400	800	ns	CxSP = 1
CM04B		Response Time Falling Edge	—	200	400	ns	CxSP = 1
CM04C		Response Time Rising Edge	—	1200	—	ns	CxSP = 0
CM04D		Response Time Falling Edge	—	550	—	ns	CxSP = 0
CM05*	TMC2OV	Comparator Mode Change to Output Valid	—	—	10	µs	
CM06	CHYSTER	Comparator Hysteresis	—	25	—	mV	CxHYS = 1, CxSP = 1

* These parameters are characterized but not tested.

Note 1: See **Section 28.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

2: Response time measured with one comparator input at VDD/2, while the other input transitions from VSS to VDD.

TABLE 27-16: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS(1)

Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	**Sym.**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
DAC01*	CLSB	Step Size	—	VDD/32	—	V	
DAC02*	CACC	Absolute Accuracy	—	—	± 1/2	LSb	
DAC03*	CR	Unit Resistor Value (R)	—	5K	—	Ω	
DAC04*	CST	Settling Time(2)	—	—	10	µs	

* These parameters are characterized but not tested.

Note 1: See **Section 28.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

2: Settling time measured while DACR<4:0> transitions from '0000' to '1111'.

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (2/2015)

Initial release of this document.

Revision B (09/2015)

Added Section 5.4: Clock Switching Before Sleep.

Updated Low-Power Features and Memory sections on cover page.

Updated Examples 3-2 and 16-1; Figures 8-1, 22-1, and 23-8 through 23-13; Registers 8-1, 23-6, 24-2, and 24-3; Sections 8.2.2, 16.2.6, 22.0, 23.3.3, 24.9.1.2, 24.11.1 and 27.1; and Tables 27-1, 27-2, 27-3, 27-8 and 27-11.

Revision C (01/2016)

Added graphs to chapter “DC and AC Characteristics Graphs and Charts”. Other minor corrections.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-5224-0190-2

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.