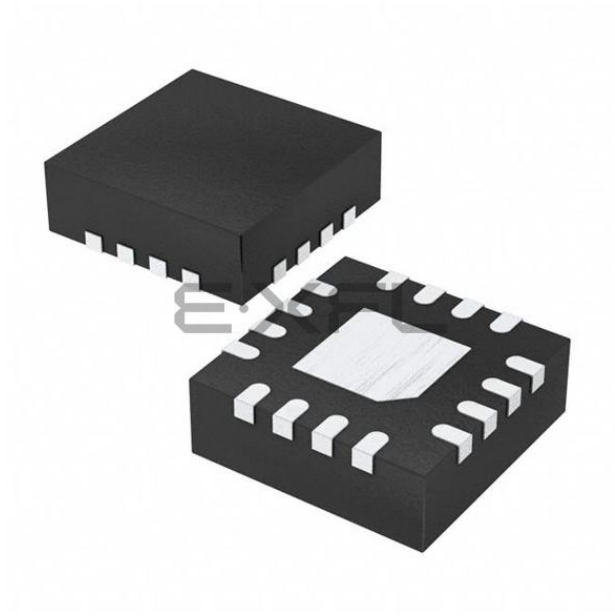


Welcome to E-XFL.COM

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	I²C, LINbus, SPI, 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 8x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-VQFN Exposed Pad
Supplier Device Package	16-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1614-i-ml

20-pin QFN, UQFN

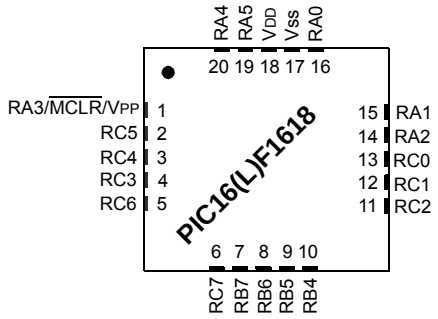


TABLE 3-14: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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
Banks 15											
78Ch to 790h	—	Unimplemented								—	—
791h	CRCDATL	DAT<7:0>								xxxx xxxx	xxxx xxxx
792h	CRCDATH	DAT<15:8>								xxxx xxxx	xxxx xxxx
793h	CRCACCL	ACC<7:0>								0000 0000	0000 0000
794h	CRCACCH	ACC<15:8>								0000 0000	0000 0000
795h	CRCSHIFTL	SHIFT<7:0>								0000 0000	0000 0000
796h	CRCSHIFTH	SHIFT<15:8>								0000 0000	0000 0000
797h	CRCXORL	XOR<7:1>							—	xxxx xxx-	xxxx xxx-
798h	CRCXORH	XOR<15:8>								xxxx xxxX	xxxx xxxX
799h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	0000 --00	0000 -00
79Ah	CRCCON1	DLEN<3:0>				PLEN<3:0>				0000 0000	0000 0000
79Bh to 79Fh	—	Unimplemented								—	—

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

Note 1: PIC16F1614/8 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1614 only.

4: PIC16(L)F1618 only.

TABLE 3-14: 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 (Continued)											
DA5h	SMT2CPWH	SMT2CPW<15:8>								xxxx xxxx	xxxx xxxx
DA6h	SMT2CPWU	SMT2CPW<23:16>								xxxx xxxx	xxxx xxxx
DA7h	SMT2PRL	SMT2PR<7:0>								xxxx xxxx	xxxx xxxx
DA8h	SMT2PRH	SMT2PR<15:8>								xxxx xxxx	xxxx xxxx
DA9h	SMT2PRU	SMT2PR<23:16>								xxxx xxxx	xxxx xxxx
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT2PS<1:0>		0-00 0000	0-00 0000
DABh	SMT2CON1	SMT2GO	REPEAT	—	—	MODE<3:0>				00-- 0000	00-- 0000
DACh	SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	000- -000	000- -000
DADh	SMT2CLK	—	—	—	—	—	CSEL<2:0>			---- -000	---- -000
DAEh	SMT2SIG	—	—	—	SSEL<4:0>					---0 0000	---0 0000
DAFh	SMT2WIN	—	—	—	WSEL<4:0>					---0 0000	---0 0000

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

Note 1: PIC16F1614/8 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1614 only.

4: PIC16(L)F1618 only.

4.3 Code Protection

Code protection allows the device to be protected from unauthorized access. Internal access to the program memory is unaffected by any code protection setting.

4.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Words. When $\overline{CP}$ = 0, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See **Section 4.4 "Write Protection"** for more information.

4.4 Write Protection

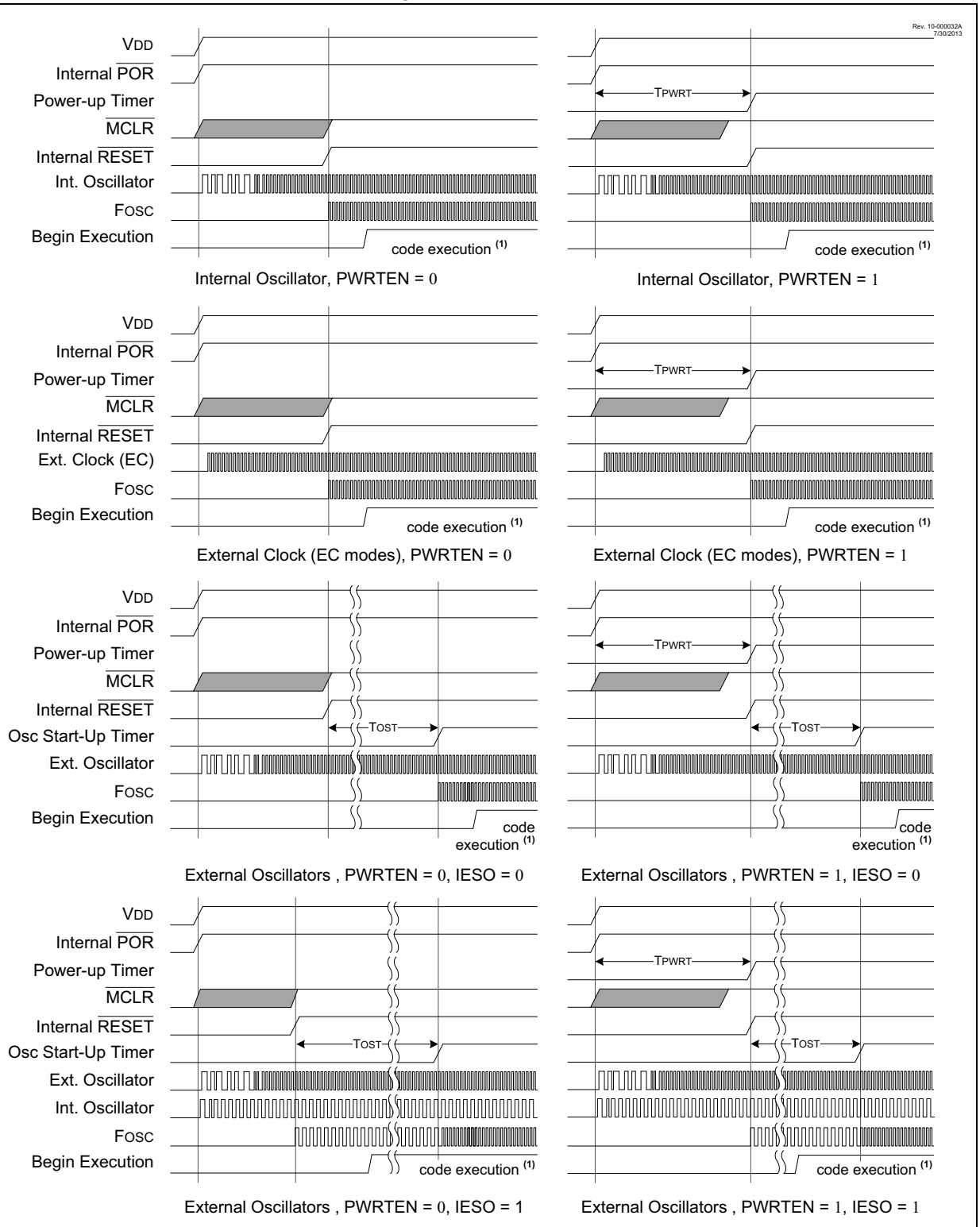
Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Words define the size of the program memory block that is protected.

4.5 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 10.4 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the **"***PIC12(L)F1612/16(L)F161X Memory Programming Specification*" (DS40001720).

FIGURE 6-3: RESET START-UP SEQUENCE



Note 1: Code execution begins 10 FOSC cycles after the FOSC clock is released.

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.

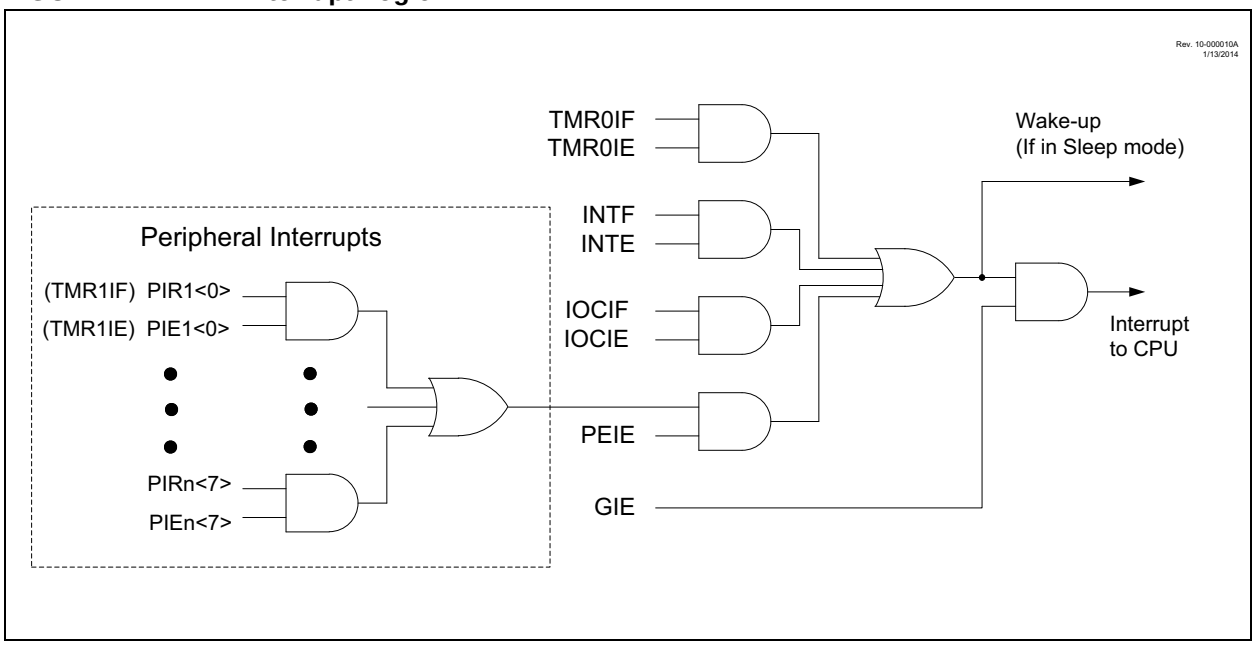
This chapter contains the following information for Interrupts:

- 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



REGISTER 11-14: SCANHADRH: SCAN HIGH ADDRESS HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
HADR<15:8>(1, 2)							
bit 7							bit 0

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

bit 7-0 **HADR<15:8>**: Scan End Address bits(1, 2)
Most Significant bits of the address at the end of the designated scan

Note 1: Registers SCANHADRH/L form a 16-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 11-15: SCANHADRL: SCAN HIGH ADDRESS LOW BYTE REGISTER

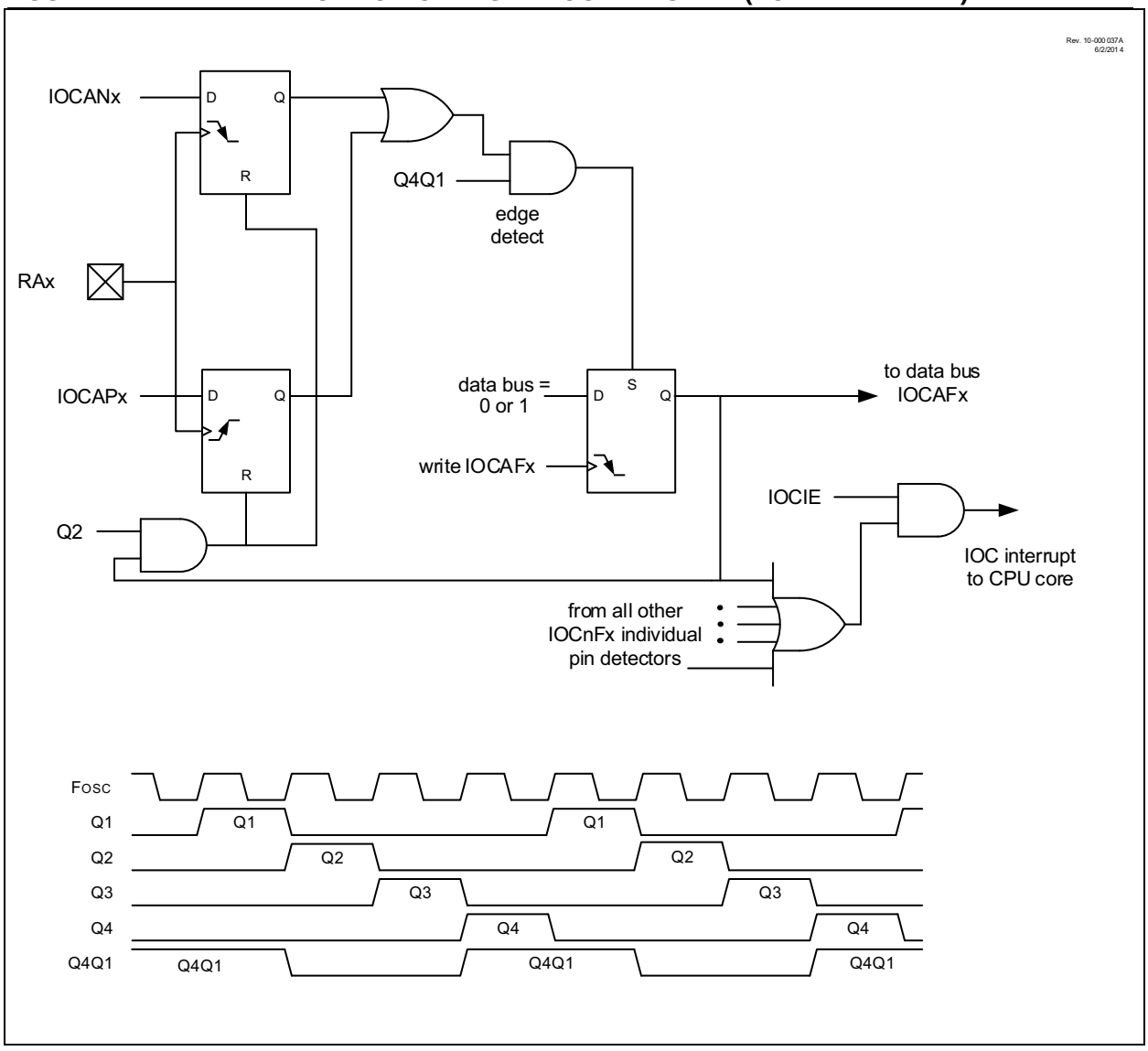
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
HADR<7:0>(1, 2)							
bit 7							bit 0

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

bit 7-0 **HADR<7:0>**: Scan End Address bits(1, 2)
Least Significant bits of the address at the end of the designated scan

Note 1: Registers SCANHADRH/L form a 16-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

FIGURE 14-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)



14.6 Register Definitions: Interrupt-on-Change Control

REGISTER 14-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

U-0	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
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAP<5:0>**: Interrupt-on-Change PORTA Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 14-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

U-0	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
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 IOCAN<5:0>: Interrupt-on-Change PORTA Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 14-3: IOCAF: INTERRUPT-ON-CHANGE PORTA FLAG REGISTER

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
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	HS - Bit is set in hardware

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAF<5:0>**: Interrupt-on-Change PORTA Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCAPx = 1 and a rising edge was detected on RAx, or when IOCANx = 1 and a falling edge was detected on RAx.
0 = No change was detected, or the user cleared the detected change.

16.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, "*Use and Calibration of the Internal Temperature Indicator*" (DS01333) for more details regarding the calibration process.

16.1 Circuit Operation

Figure 16-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 16-1 describes the output characteristics of the temperature indicator.

EQUATION 16-1: V_{OUT} RANGES

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

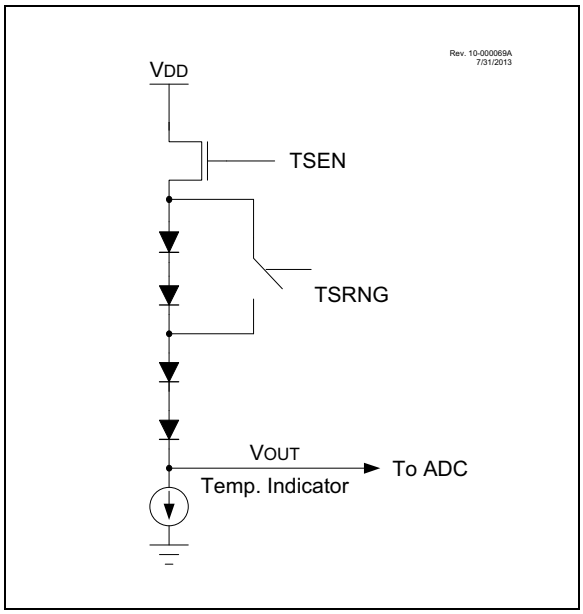
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section15.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher V_{DD} is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 16-1: TEMPERATURE CIRCUIT DIAGRAM



16.2 Minimum Operating V_{DD}

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, V_{DD}, must be high enough to ensure that the temperature circuit is correctly biased.

Table 16-1 shows the recommended minimum V_{DD} vs. range setting.

TABLE 16-1: RECOMMENDED V_{DD} VS. RANGE

Min. V_{DD}, TSRNG = 1	Min. V_{DD}, TSRNG = 0
3.6V	1.8V

16.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section17.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

16.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 μs after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 μs between sequential conversions of the temperature indicator output.

17.2 ADC Operation

17.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section17.2.6 "ADC Conversion Procedure"**.

17.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

17.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

17.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. Performing the ADC conversion during Sleep can reduce system noise. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

17.2.5 AUTO-CONVERSION TRIGGER

The auto-conversion trigger allows periodic ADC measurements without software intervention. When a rising edge of the selected source occurs, the GO/$\overline{\text{DONE}}$ bit is set by hardware.

The auto-conversion trigger source is selected with the TRIGSEL<4:0> bits of the ADCON2 register.

Using the auto-conversion trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See Table 17-2 for auto-conversion sources.

TABLE 17-2: AUTO-CONVERSION SOURCES

Source Peripheral	Signal Name
Timer0	T0_overflow
Timer1	T1_overflow
Timer2	TMR2_postscaled
Timer4	TMR4_postscaled
Timer6	TMR6_postscaled
Comparator C1	C1_OUT_sync
Comparator C2	C2_OUT_sync
SMT1	SMT1_CPW
SMT1	SMT1_CPR
SMT1	SMT1_PR
SMT2	SMT2_CPW
SMT2	SMT2_CPR
SMT2	SMT2_PR
CCP1	CCP1_out
CCP2	CCP2_out

The pull-up and pull-down resistor values are significantly affected by small variations of VCPINV. Measuring VCPINV can be difficult, especially when the waveform is relative to VDD. However, by combining Equations 20-2 and 20-3, the resistor value can be determined from the time difference between the ZCDx_output high and low periods. Note that the time difference, ΔT, is 4*TOFFSET. The equation for determining the pull-up and pull-down resistor values from the high and low ZCDx_output periods is shown in Equation 20-4. The ZCDx_output signal can be directly observed on the ZCDxOUT pin by setting the ZCDxOE bit.

EQUATION 20-4:

$$R = R_{SERIES}\left(\frac{V_{BIAS}}{V_{PEAK}\left(\sin\left(\pi Freq\frac{(\Delta T)}{2}\right)\right)} - 1\right)$$

R is pull-up or pull-down resistor.

VBIAS is VPULLUP when R is pull-up or VDD when R is pull-down.

ΔT is the ZCDxOUT high and low period difference.

20.6 Handling VPEAK variations

If the peak amplitude of the external voltage is expected to vary, the series resistor must be selected to keep the ZCD current source and sink below the design maximum range of ± 600 μA and above a reasonable minimum range. A general rule of thumb is that the maximum peak voltage can be no more than six times the minimum peak voltage. To ensure that the maximum current does not exceed ± 600 μA and the minimum is at least ± 100 μA, compute the series resistance as shown in Equation 20-5. The compensating pull-up for this series resistance can be determined with Equation 20-3 because the pull-up value is independent from the peak voltage.

EQUATION 20-5: SERIES R FOR V RANGE

$$R_{SERIES} = \frac{V_{MAXPEAK} + V_{MINPEAK}}{7\times10^{-4}}$$

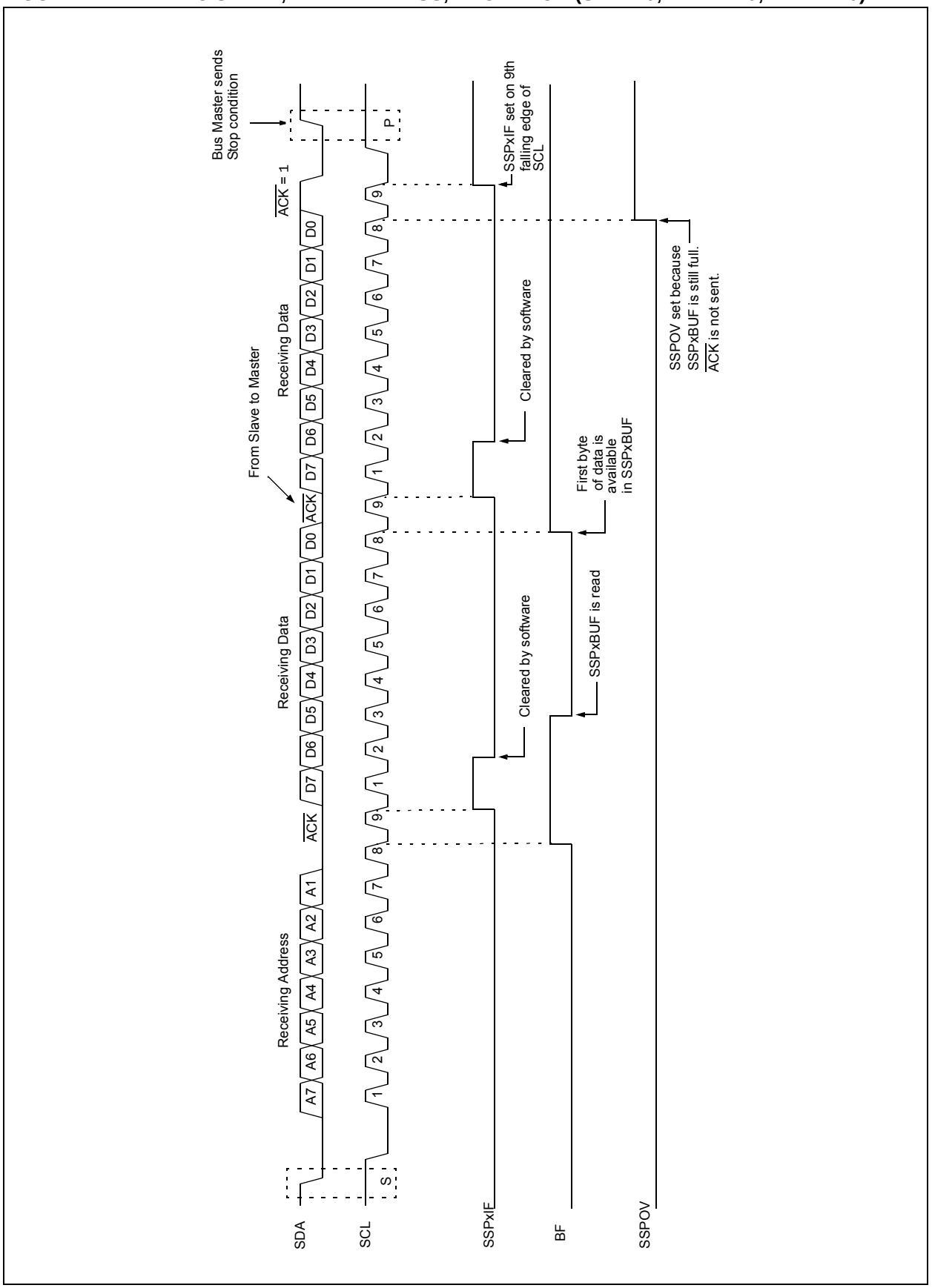
20.7 Operation During Sleep

The ZCD current sources and interrupts are unaffected by Sleep.

20.8 Effects of a Reset

The ZCD circuit can be configured to default to the active or inactive state on Power-On-Reset (POR). When the $\overline{ZCD}$ Configuration bit is cleared, the ZCD circuit will be active at POR. When the $\overline{ZCD}$ Configuration bit is set, the ZCDxEN bit of the ZCDxCON register must be set to enable the ZCD module.

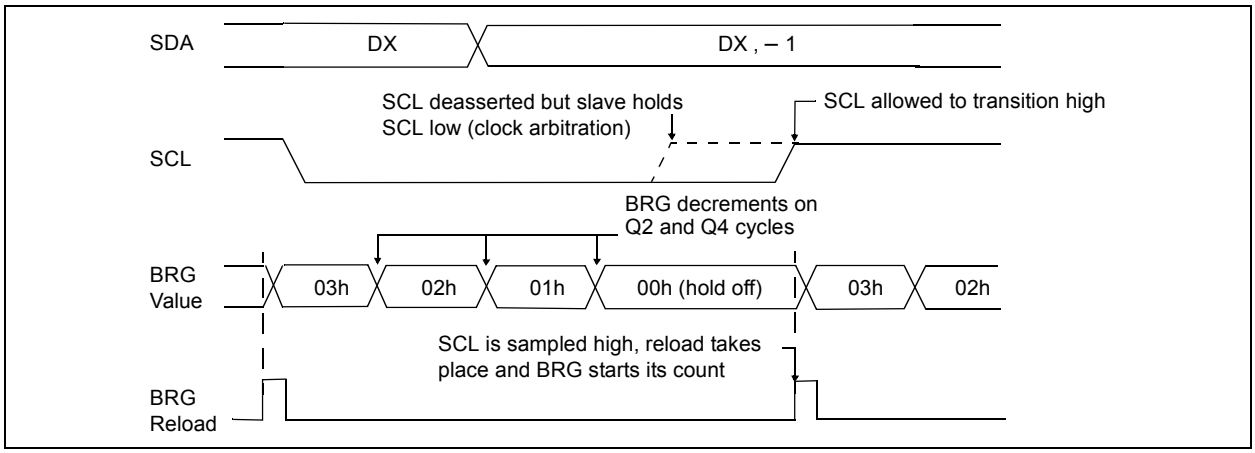
FIGURE 24-14: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)



24.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 24-25).

FIGURE 24-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



24.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not idle.

Note:	Because queuing of events is not allowed, writing to the lower five bits of SSPxCON2 is disabled until the Start condition is complete.

28.1.4 STEERING MODES

In Steering modes, the data input can be steered to any or all of the four CWG output pins. In Synchronous Steering mode, changes to steering selection registers take effect on the next rising input.

In Non-Synchronous mode, steering takes effect on the next instruction cycle. Additional details are provided in **Section 28.9 “CWG Steering Mode”**.

FIGURE 30-3: TIMER MODE TIMING DIAGRAM

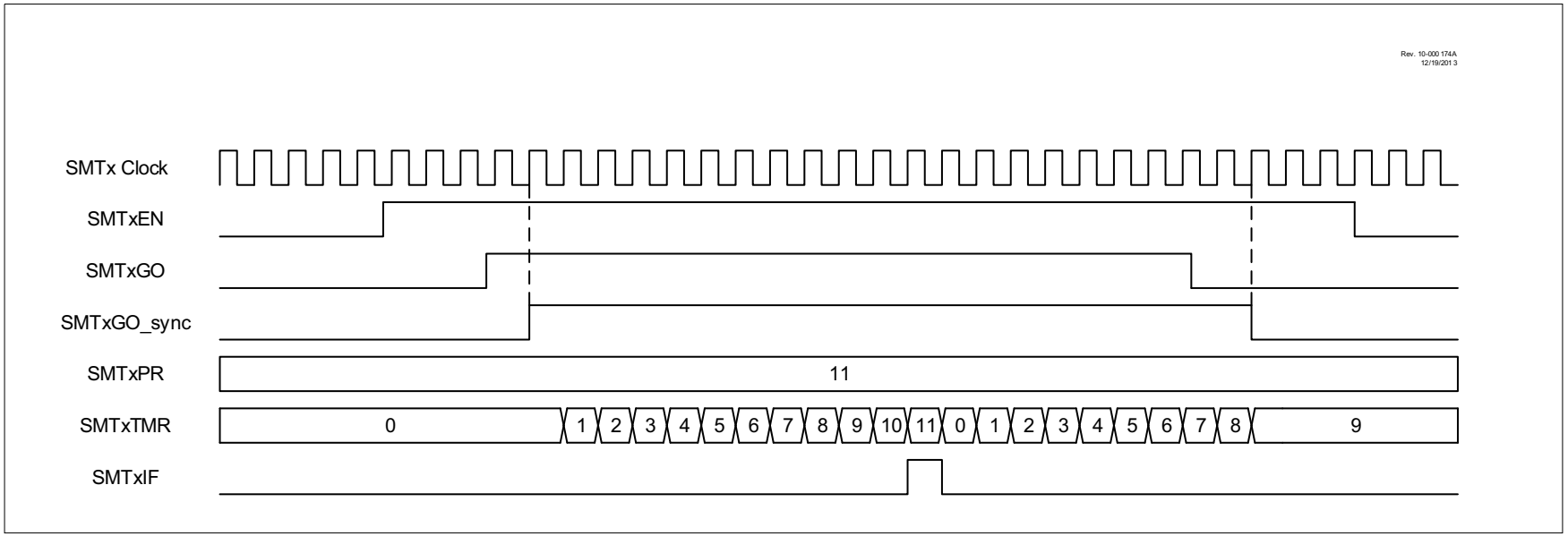
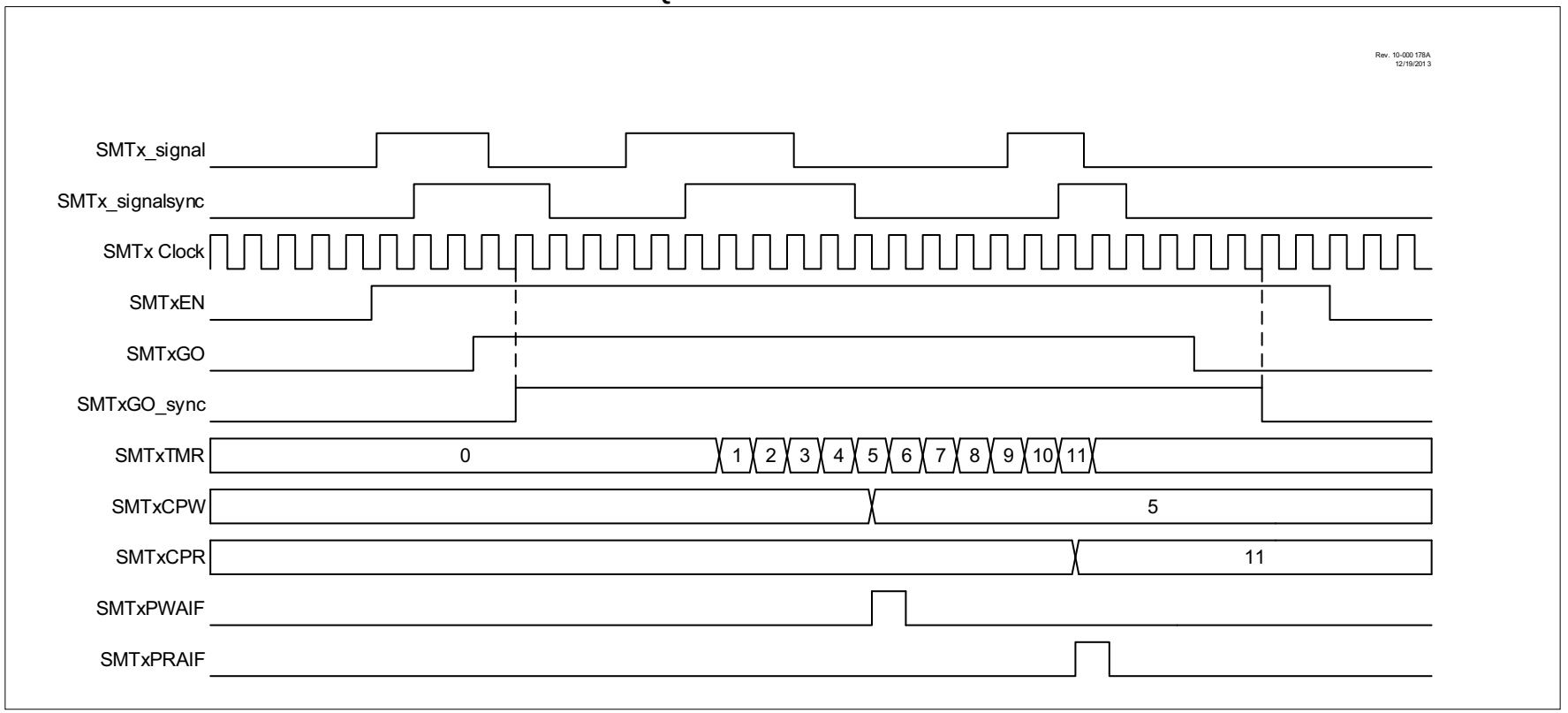


FIGURE 30-7: PERIOD AND DUTY-CYCLE SINGLE ACQUISITION TIMING DIAGRAM



31.0 ANGULAR TIMER (AT) MODULE

The Angular Timer (AT) module subdivides periodic signals into smaller equally spaced intervals, the number of which remain constant as the periodic signal frequency changes. A counter tracks the intervals starting at zero at each period event. The counter can be compared to user defined values to cause events, or the counter value can be captured by events external to the module. This allows for a variety of applications, such as measuring of A/C mains, stall detection for motors, commutation for brushless motors, and TDC detection for internal combustion engines. A second counter tracks the period time. This can be used to measure the error of the period based on a pre-programmed set point, as well as detect missing pulses in the signal. The angular timer includes the following features:

- Two operating modes
  - Single-pulse per period
  - Multiple-pulses per period
- Two missing pulse modes
  - Adaptive
  - Fixed
- Multiple selectable clock sources
- Phase clock output with polarity control
- Period clock output with polarity control
- Missing pulse output with polarity control
- Interrupts for phase and period clock generation, as well as for missing pulse detect
- Period set point and error register
- Compare-pulse outputs
  - Independent interrupts
- Capture inputs
  - Input polarity control
  - Independent interrupts

31.1 Principle of Operation

Consider the statements in Equation 31-1:

EQUATION 31-1:

If:	And:	Then:
$P = \frac{F}{R}$	$A = \frac{F}{P}$	$A = R$

In these three equations:

- P represents the period count ATxPER
- A represents the angle or phase count ATxPHS
- R represents the desired resolution ATxRES
- F represents some arbitrary scaler value

Notice that the phase count equals the desired resolution regardless of what F is. If we let F equal the ratio of a system clock to the input signal frequency then that means the phase count is a constant equaling the desired resolution regardless of the input frequency. This has many extraordinary uses including:

- Use phase compare feature to create an event at a fixed phase angle in the period
- Use capture feature to capture the phase angle at which an event occurs
- Use error feature to monitor deviations from a user specified period time

The details of these features, and more, are described in the following sections of this chapter.

31.2 Angular Timer Operating Modes

The AT module operates in two basic modes:

- Single-Pulse mode
- Multi-Pulse mode

Both modes function on the same principle: Dividing a periodic input signal into intervals, and allowing events to trigger off of these smaller intervals. The primary difference between these two modes is how the period is determined. The Single-Pulse mode determines the period as the time between every pulse in the input pulse stream. The Multi-Pulse mode determines the period as the time between missing pulses in the input pulse stream.

The primary parameter for both modes is the ATxRES register pair. This value is used to determine the granularity of the phase counter and the frequency of the phase clock output of the module.

MOVLP **Move literal to PCLATH**

Syntax: [*label*] MOVLP k

Operands: $0 \le k \le 127$

Operation: $k \rightarrow PCLATH$

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW **Move literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow (W)$

Status Affected: None

Description: The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

Words: 1

Cycles: 1

Example:

```
MOVLW   0x5A
```

After Instruction

W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: $(W) \rightarrow (f)$

Status Affected: None

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example:

```
MOVWF   OPTION_REG
```

Before Instruction

OPTION_REG = 0xFF

W = 0x4F

After Instruction

OPTION_REG = 0x4F

W = 0x4F

MOVWI **Move W to INDFn**

Syntax:

[*label*] MOVWI ++FSRn
[*label*] MOVWI --FSRn
[*label*] MOVWI FSRn++
[*label*] MOVWI FSRn--
[*label*] MOVWI k[FSRn]

Operands:

$n \in [0,1]$
$mm \in [00,01, 10, 11]$
$-32 \le k \le 31$

Operation: $W \rightarrow INDFn$

Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected: None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

XORLW	Exclusive OR literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k $\rightarrow$ (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ (destination)
Status Affected:	Z
Description:	Exclusive OR 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'.